

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК 004.043

«До захисту допущено»

Завідувач кафедри
_____ І.Р. Пархомей
(підпис)

“ ____ ” _____ 2019 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 121 «Інженерія програмного забезпечення»

на тему: Система розпізнавання домашніх тварин для розумного дому

Виконав: студент другого курсу, групи ІТ-84мп
(шифр групи)

_____ Паламарчук Ігор Олегович
(прізвище, ім'я, по батькові)

_____ (підпис)

Науковий керівник доцент, к.т.н., доцент Корнага Я.І.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант _____
(назва розділу) (науковий ступінь, вчене звання, прізвище, ініціали)

_____ (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ І.Р. Пархомей
(підпис)

«___» _____ 2019 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Паламарчуку Ігорю Олеговичу
(прізвище, ім'я, по батькові)

1. Тема дисертації «Система розпізнавання домашніх тварин для розумного дому», _____

науковий керівник дисертації доцент, к.т.н., доцент Корнага Я. І., _____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «___» _____ 2019 р. № _____

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження – системи розпізнавання об'єктів.

4. Предмет дослідження – алгоритми за допомогою яких можна розпізнавати об'єкти.

5. Перелік завдань, які потрібно розробити – аналіз проблеми та існуючих рішень; аналіз і реалізація методу; розробка методу; розробка програмного забезпечення; дослідження ефективності розробленого методу.

6. Орієнтовний перелік ілюстративного матеріалу – шість плакатів

7. Орієнтовний перелік публікацій – одна публікація

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області	13.09.2019 р.	
2	Постановка задачі	15.09.2019 р.	
3	Аналіз інформаційного забезпечення	20.09.2019 р.	
5	Аналіз алгоритмічного забезпечення	25.09.2019 р.	
6	Розробка алгоритмічного забезпечення	15.10.2019 р.	
7	Розробка програмного забезпечення	01.11.2019 р.	
8	Маркетинговий аналіз стартап-проекту	10.11.2019 р.	
9	Висновки	15.11.2019 р.	

Студент

_____ (підпис)

Т. В. Оксенчук

(ініціали, прізвище)

Науковий керівник дисертації

_____ (підпис)

Я. І. Корнага

(ініціали, прізвище)

АНОТАЦІЯ

З кожним роком зростає зацікавленість у вирішенні складних задач розпізнавання об'єктів, що обумовлена автоматизацією, необхідністю образних процесів комунікації в інтелектуальних системах. Тому удосконалення реалізації розпізнавання комп'ютерними системами образів є актуальною. Один з перспективних напрямків вирішення даної проблеми ґрунтується на застосуванні штучних нейронних мереж і нейрокомп'ютерів, як найбільш прогресивних по відношенню до проблем класифікації задач розпізнавання образів. У наш час запропоновано велику кількість архітектур нейромреж для застосування у розпізнаванні об'єктів. Аналіз запропонованих рішень показує, що й досі не існує такої моделі, яка б була кращою серед усіх результируючих показників роботи. Перспективу в удосконаленні архітектур вбачають у згорткових нейронних мережах. Переваги згорткових мереж над багатопшаровими полягають у використанні спільної ваги у згорткових шарах, що означає, що для кожного пікселя шару використовується один і той же фільтр (банк ваги).

У роботі розглянуто проблему автоматизованого розпізнавання домашніх тварин, показано основні особливості існуючих рішень та додатків, їх переваги та недоліки. Для визначення тварин розроблена нейронна мережа, що розпізнає об'єкти на зображенні. Визначено завдання для системи розпізнавання домашніх тварин на зображенні за допомогою нейронної мережі та відібрано нейронну мережу та спосіб навчання, які найбільш підходять для даної задачі. Описано структуру нейронної мережі та проведено експерименти по навчанню та її роботі.

Ключові слова: машинне навчання, нейронна мережа, розпізнавання, інтелектуальна система.

Розмір пояснювальної записки – 75 аркушів, містить 28 ілюстрації, 25 таблиць, 6 додатків.

ABSTRACT

The Relevance. Year by year, the interest in solving more complex tasks of object recognition is growing, due to automation needs for shaped communication processes in intelligent systems. Therefore, improving the implementation of the recognition of computer image systems is relevant. One of the promising directions for solving this problem is based on the use of artificial neural networks and neurocomputers as the most progressive in relation to the problems of classification of pattern recognition tasks. In our time, a large number of neural network architectures are proposed for application in the recognition of objects. The analysis of the proposed solutions shows that there is still no such model that would be the best among all the resulting performance parameters. Prospects for the improvement of architecture are seen in convolutional neural networks. The advantages of roller networks over multilayers are to use a common weight in the roller coasters, which means that for each pixel of the layer is used the same filter (weight).

The paper considers with the problem in the field of automated recognition of pets, shows the main features of existing solutions and applications, their advantages and disadvantages. A neural network has been developed that recognize and identify the animals on the image. In this work has been determined task for the recognition system to recognize and identify pets on the pictures with the help of the neural networks. Also, the neural network was trained. The tasks for the recognition system of domestic animals in the picture using the neural network are determined and the neural network and the training method are selected, are most suitable for this task. The structure of the neural network is described and experiments for its training are carried out.

Keywords: machine learning, neural network, recognition, intelligent system.

The work is presented on 75 pages, contains 28 images, 25 tables and 6 additions.

**Пояснювальна записка
до магістерської дисертації**

на тему: *Система розпізнавання домашніх тварин для розумного
дому*

Київ – 2019 року

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ	9
ВСТУП.....	11
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ...	13
1.1 Об'єкт та предмет дослідження.....	13
1.2 Машинне навчання	13
1.3 Історія розвитку нейронних мереж	15
1.4 Поняття «штучна нейронна мережа»	17
1.4.1 Біологічний та штучний нейрони.....	18
1.4.2 Класифікації штучних інтелектуальних систем	21
1.5 Аналіз існуючих рішень	27
1.5.1 Аналіз системи «Google Lens».....	27
1.5.2 Аналіз системи «Image Recognizer»	28
1.5.3 Аналіз системи «Imagga»	30
1.6 Порівняння існуючих програмних рішень	31
1.7 Постановка задачі.....	32
Висновки до розділу	34
РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ	35
2.1 Типи нейронних мереж.....	35
2.1.1 Одношарові штучні нейронні мережі.	35
2.1.2 Багатошарові штучні нейронні мережі.	36
2.1.3 Рекурентні нейронні мережі	37
2.1.4 Нейронні мережі прямого поширення та перцептрон	39
2.1.5 Нейронна мережа Гопфілда	40
2.1.6 Ланцюги Маркова	41
2.1.7 Машина Больцмана.....	42
2.1.8 Згорткові нейронні мережі	43
2.2 Вибір нейронної мережі для розпізнавання тварин на фото	45
2.2.1 Алгоритм роботи згорткової нейромережі.....	45
2.2.2 Основні види навчання згорткових нейромереж.....	46
Висновки до розділу	48

РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	49
3.1 Вибір архітектури та платформи розробки	49
3.1.1 Трирівнева архітектура.....	49
3.1.2 Платформа Microsoft .NET Core 3.0	51
3.1.3 Мова програмування C#.....	52
3.1.4 СУБД Microsoft SQL Server 2019	52
3.1.5 Технологія Entity Framework Core.....	53
3.1.6 Бібліотека ML.NET	55
3.2 Реалізація серверної частини та навчання нейромережі.....	56
3.3 Використання системи та алгоритм її роботи.....	59
Висновки до розділу	62
РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ	63
4.1 Опис ідеї проекту	63
4.2 Технологічний аудит ідеї проекту.....	65
4.3 Аналіз ринкових можливостей запуску стартап-проекту.....	66
4.4 Розроблення ринкової стратегії проекту	74
4.5 Розроблення маркетингової програми стартап-проекту.....	76
Висновки до розділу	79
ВИСНОВКИ.....	80
ПЕРЕЛІК ПОСИЛАНЬ	81
ДОДАТОК А.....	83
ДОДАТОК Б	84
ДОДАТОК В.....	86

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

ЗНМ – згорткова нейронна мережа;

БД – база даних;

БН – біологічний нейрон;

EF – Entity Framework;

ML – Machine Learning;

WPF – Windows Presentation Foundation;

ОС – операційна система;

СУБД – Система управління базами даних;

ПЗ – Програмне забезпечення;

ІТ – інформаційні технології.

IIS – Internet Information Server;

SQL – Structured query language;

Framework – інфраструктура програмних рішень, що полегшує розробку складних систем;

LINQ – Language Integrated Query;

Soft Computing – м'які обчислення;

API (application programming interface) - це набір чітко визначених методів для взаємодії різних компонентів;

CNN (convolutional neural networks) – згорткові нейронні мережі;

DCNN (deep convolutional neural networks) – глибокі згорткові нейронні мережі;

BM (Boltzmann machines) – машини Больцмана;

MC – (Markov chains) – ланцюги Маркова;

FFNN (feed forward neural networks) – нейронні мережі прямого поширення;

Кластер (англ. cluster) – група однакових або подібних елементів, зібраних разом або близько розташованих один до одного;

Набір даних (англ. data set) – колекція однотипних даних, що застосовується в задачах машинної обробки даних;

Синапс нейрона (від грецького «synapsis» – з'єднання) – структура, яка дозволяє нейрону (нейронної мережі) проводити сигнал (у випадку штучного нейрону) до іншого нейрону;

Аксон – слугує для передачі імпульсу нейрону;

Дендрит – такий собі приймач імпульсів, які надходять від нейронів;

ВСТУП

В нинішній час, люди все частіше і частіше звертаються за допомогою у вирішенні складних задач до систем, які використовують нейронні мережі. Це обумовлено тим, що існує велика множина завдань в різних сферах діяльності людини, наприклад, таких як: медицина (виявлення вад, наприклад – злоякісних пухлин на рентгенівських знімках), економіка (прогнозування курсу валют), робототехніка, сільське господарство (пошук правильної послідовності генів, що дасть змогу рослині, отримати ті чи інші корисні характеристики), машинобудування тощо. Зважаючи на велику кількість галузей, де застосовують машинне навчання та нейронні мережі, можна сказати, що ці системи, являють собою набір інструментів, що допомагає у вирішенні питань аналізу великих масивів даних. Без використання цих мереж, потрібно було б задіяти велику кількість людських ресурсів.

Також, нейронні мережі та машинне навчання використовують у системах розумного дому. В даній галузі вони допомагають користувачам автоматизувати процеси, які відбуваються всередині їхніх домівок. Наприклад:

- деактивація розеток, коли вони неактивні;
- увімкнення освітлення в домі, коли користувач повертається додому;
- вмикання та вимикання опалювання в певні періоди на протязі дня – система навчається визначати чи є хтось в домі, якщо немає – вимикає опалення. Також вона запам'ятовує періоди, коли хтось знаходиться у домі та вмикає опалення до приходу користувачів додому.

Існує багато людей, які обожнюють та тримають у домівках домашніх улюбленців. Зазвичай, вони змушені цілими днями сидіти вдома та сумувати, бо їх власники проводять час на роботі, або прийшовши з вулиці сидіти під дверима дому, та очікувати на своїх власників, щоб вони впустили їх всередину.

Ціллю даної роботи є розробка програмного забезпечення, що дозволить вирішити проблему розпізнавання та допуску до приміщень домашніх улюбленців в системі розумного дому з використанням технологій машинного навчання та нейронних мереж.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Об'єкт та предмет дослідження

Об'єктом дослідження являються інтелектуальні системи розпізнавання предметів на зображеннях, що використовують для цього машинне навчання та нейронні мережі.

Предметом дослідження є аналіз різних нейронних мереж, які дозволяють розпізнавати об'єкти на фото або на картинках.

Метою дослідження є визначення того, який з типів нейронних мереж краще підходить для вирішення поставленої задачі, також навчання цієї системи розпізнавати домашніх тварин в системі розумного дому.

Для того, щоб приступити до роботи, потрібно розібратися в тому, що ж таке машинне навчання, нейрон, нейронні мережі.

1.2 Машинне навчання

Машинне навчання – це галузь інформатики та комп'ютерних наук, що використовує статистичні методи, для того, щоб дати можливість комп'ютерним системам “вчитися” на тестових даних, без явного програмування алгоритмів, для вирішення подальших задач без втручання людини.

Також, машинне навчання використовують здебільшого для вирішення складних проблем, та проблем, які потребують адаптації. Можна сказати, що це такий клас завдань, який неможливо вирішити якимось певним, чітким алгоритмом, та при цьому потрібно зважати на вже отримані дані, результати. Ось декілька прикладів таких задач:

Завдання, які виконуються людиною або твариною: є безліч задач, які ми виконуємо регулярно, але аналіз того, як ми їх виконуємо недостатньо продуманий для того, щоб визначити чіткий алгоритм цих дій. Серед таких задач, можна виділити водіння, розпізнавання образів або мови тощо. Усі програми, що використовують машинне навчання для вирішення

подібних задач, досягають непоганих результатів, коли навчаються на великій кількості тренувальних даних.

Завдання, які виходять за межі людських можливостей:

це ще один широкий клас задач, що отримують користь від використання машинного навчання. Вони тісно пов'язані з аналізом та використанням великих та складних масивів даних. Це можуть бути: астрономічні дані, дані для прогнозування курсу валют, двигун для пошуку інформації у веб, перенесення медичних даних, архівів у медичні знання тощо. Серед великої кількості цієї інформації є така, що є важливою для людини, але людина не здатна самотійно віднайти цю інформацію, бо даних надто багато і зазвичай ці дані дуже складні. Однією з перспективних областей є вміння визначати значущі моделі у великих та складних наборах даних, яке в поєднанні з програмами, що навчаються з необмеженою кількістю пам'яті та постійно зростаючою швидкістю обробки даних, сильно полегшує роботу з даним типом задач.

Тож, метою машинного навчання є передбачення результату за вхідними даними. Чим різноманітнішими будуть вхідні дані, тим простіше машині знайти закономірності і тим точнішим буде результат.

Для того, щоб навчити машину нам потрібно 3 речі:

1. Дані. Для того, щоб виявити спам – потрібно мати приклади спам-листів, щоб передбачити курс акцій – потрібно мати історію цін, щоб дізнатися інтереси користувача – потрібні його пости у соціальних мережах та його лайки. Даних потрібно чим більше, бо чим їх буде більше – тим кращим буде фінальний результат. Дані збирають по-різному, дехто збирає їх вручну – це тривалий процес, але хоч даних менше, зате ці дані будуть без помилок. Інші збирають дані в автоматичному режимі – віддають машині всю знайдену інформацію та розраховують на хороший результат. Найхитріші – компанії, типу Google, використовують своїх користувачів для навчання подібних систем. Згадайте ту ж саму ReCaptcha-у, що вимагає від користувача вказати на картинки, на яких зображено дорожні знаки, автомобілі, тощо.

2. Ознаки. Їх ще називають фічами (features). Ознаки, властивості, характеристики – це може бути що завгодно – ціна акцій, стать користувача, пробіг мотоцикла, навіть той же ітератор частоти появи певного слова в тексті. Машина повинна конкретно знати, на що їй дивитися, що шукати. Добре мати потрібні дані, наприклад в табличках в базі даних, де назви колонок і будуть фічами, тобто ознаками чи характеристиками. А що, якщо у нас є п'ятдесят гігабайтів картинок з собаками, які мають велику кількість характеристик? У випадку, коли ознак надто багато, модель працюватиме дуже повільно та неефективно. Відбір правильних ознак, найчастіше займає більшу кількість часу при навчанні подібних систем. Також, бувають зворотні ситуації, коли користувач вирішує самотужки вибрати "правильні" на його думку характеристики, що призводить до того, що система починає неправильно працювати та помилятися.

3. Алгоритм. Зазвичай, одну й ту ж саму задачу можна вирішити різними методами або способами. Від вибору методу буде залежати швидкість, точність роботи та головне – розмір готової моделі. Але завжди потрібно зважати на одну річ: якщо тестові дані – "сміття", то в даному випадку ніякий алгоритм не допоможе. Чим більше буде "правильних" тестових даних, тим більшою буде імовірність отримати задовільний результат.

Тож, що таке машинне навчання ми розібралися. Переходимо до нейронних мереж та нейронів.

1.3 Історія розвитку нейронних мереж

Вперше про штучні нейронні мережі заговорили в 1940-х роках. Саме як наукова дисципліна теорія нейронних мереж була відображена в роботі Мак Каллока і Пітса в 1943 році [1]. У даній роботі стверджувалося, що майже будь-яку логічну або арифметичну функцію можна реалізувати за допомогою найпростішої нейронної мережі.

У 1949 році Дональд Гебб сформував закон, який став відправним пунктом для навчання нейронних мереж. Він припустив, що навчання, перше за все

полягає в зміні сили синаптичних зв'язків. Його теорія – типовий випадок самонавчання, де випробувана система навчається виконувати потрібну задачу без втручання експериментатора. Свій внесок зробив і Марвін Мінський, завдяки дослідженню ряду задач, в тому числі відомої задачі виключної диз'юнкції [2, с.7].

Ф. Розенблат в 1958 році запропонував нейронну мережу, названу перцептроном, який був призначений для класифікації об'єктів. При навчанні перцептрон отримував повідомлення від «вчителя». Завдяки повідомленню, можна визначити до якого класу належить даний об'єкт. Крім того, навчений перцептрон був здатний самостійно класифікувати об'єкти, які раніше не використовувались, роблячи при цьому мізерну кількість помилок.

Період затишшя у розвитку нейронних мереж припав на 1968-1985рр. З появою високопродуктивних персональних комп'ютерів з'явилася можливість моделювати нейронні мережі. Настільною книгою фахівців, які цікавились теорією нейронних мереж, стала робота Філіпа Уоссермена «нейрокомп'ютерна техніка».

Коли з'явилася робота Джона Гопфілда у 1982 році, зацікавленість до нейронних мереж сильно зросла. Гопфілд, ґрунтуючись на правилах навчання Гебба, показав, що задачі з нейронами можуть бути зведені до узагальненого ряду моделей, розроблених на той час у фізиці неупорядкованих систем.

У 80 роках поступово сформувався міцний теоретичний фундамент, на основі якого сьогодні створюється більшість мереж. Розроблена теорія стала широко застосовуватися в останні два десятиліття для вирішення прикладних завдань. Почали з'являтися компанії, що займаються розробкою програмного забезпечення для конструювання штучних нейронних мереж.

У 90-ті роки нейронні мережі стали використовувати в бізнесі, де вони показали колосальну ефективність при вирішенні багатьох завдань – від передбачення попиту на продукцію до аналізу платоспроможності клієнтів банку.

Джефрі Гінтон у 2007 році створив алгоритми глибокого навчання нейронних мереж. При навчанні нижніх слоїв мережі, Гінтон використовував обмежену машину Больцмана, яка являє собою стохастичну рекурентну нейронну мережу. Після навчання мережі, отриманий застосунок міг швидко виконувати поставлену задачу (наприклад, пошук лиць на фото). Дану функцію вмонтовано у всі цифрові фотоапарати. Подібна технологія використовується пошуковими системами в інтернеті для класифікації картинок [3].

За оцінками фахівців, в області проектування нейронних мереж і нейрокомп'ютерів очікується технологічний ріст. Чимало нових можливостей було відкрито за останні роки, а праці в даній області стають важливим внеском в науку, технології, економіку. Не зважаючи на те, що вивчення нейронного моделювання триває вже понад шістдесят років, немає жодної області мозку, де процес обробки інформації був би зрозумілий до кінця. Також немає жодного нейрона, для якого можна було б визначити код передачі інформації у вигляді послідовності імпульсів.

В даний час існує велика кількість конфігурацій нейронних мереж, які відрізняються за принципами функціонування, а отже, спрямовані на виконання різних задач.

Майбутнє нейрокомп'ютерних технологій буде пов'язано з новими відкриттями в області нейронного моделювання – як тільки вдасться розгадати таємницю функціонування хоча б однієї області мозку, відразу ж стане багато чого зрозуміло і про інші його області.

1.4 Поняття «штучна нейронна мережа»

Можна сказати, що штучна нейронна мережа являється математичною моделлю, а також її програмною та апаратною реалізацією, що побудована за принципом, схожим до біологічних нейронних мереж – нервових клітин живого організму. Дане поняття з'явилося при спробі змодельовати процеси, що відбуваються у мозку людини [4, с.3].

Дана мережа представляє систему, яка складається з простих процесорів, що з'єднанні та функціонують між собою. Кожен з процесорів системи має справу з сигналами, які періодично надходять або передаються іншими процесорами. Велика мережа здатна вирішувати дуже складні задачі за короткий період часу.

З математичної точки зору нейронні мережі являють собою спосіб вирішення нелінійних задач оптимізації. У кібернетиці використовується теорія нейронних мереж для вирішення задач адаптивного управління, побудови алгоритмів для робототехніки [1].

Програмування нейронних мереж являє собою саме навчання мережі, а не написання програмного коду. За допомогою навчання система може визначати залежності між даними (вхідними і вихідними), узагальнювати, спрощувати та використовувати знання для розбиття важких задач на більш легкі та малі.

1.4.1 Біологічний та штучний нейрони

Людський мозок та нервова система складаються з нейронів, що з'єднані між собою нервовими волокнами. За допомогою цих волокон передаються імпульси між нейронами. Все, що відбувається з організмом, наприклад процес мислення – це взаємодія нейронів один з одним. Ось так виглядає будова біологічного нейрона – рис. 1.1 [4].

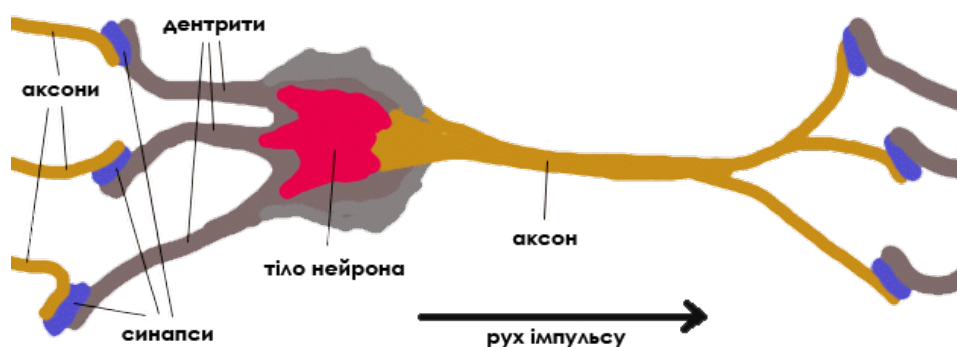


Рисунок 1.1 – Вигляд біологічного нейрона [5]

Синапс – це таке утворення, яке має вплив на силу імпульсу, для забезпечення контакту між аксоном та дендритом.

Аксон – слугує для передачі імпульсу нейронові.

Дендрит – такий собі приймач імпульсів, які надходять від нейронів.

Коли імпульс проходить через синапс, його сила змінюється в певну кількість разів, цей процес називається вагою синапса. Коли до нейрона надходять імпульси за декількома дендритами, то вони сумуються. Нейрон переходить в стан збудження і формує свій власний імпульс, який буде передано далі аксону, якщо у сумарного імпульсу перевищено поріг. Зважаючи на те, що вага синапсів може змінюватися з часом, разом із нею змінюється поведінка самого нейрона. Ось так виглядає математична модель описаного вище процесу - рис. 1. 2.

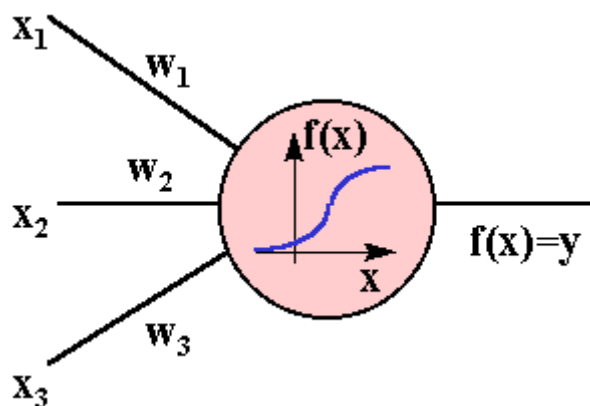


Рисунок 1.2 – Математична модель описаного процесу [5]

На малюнку зображено модель нейрону з трьома входами (дендритами), де синапси мають такі ваги w_1, w_2, w_3 , до яких надходять сили x_1, x_2, x_3 відповідно. До нейрону надходять імпульси $x_1 w_1, x_2 w_2, x_3 w_3$ після проходження синапсів та дендритів.

Отриманий сумарний імпульс $x = x_1 w_1 + x_2 w_2 + x_3 w_3$ нейрон перетворить у відповідності до передатної функції $f(x)$.

$y = f(x) = f(x_1 w_1 + x_2 w_2 + x_3 w_3)$ – сила вихідного імпульсу. Тож, таким чином, нейрон описується своїми вагами та передатною функцією. В кінцевому результаті отримуємо набір чисел x_k у вигляді входів. Далі нейрон видає деяке число y на вихід[6].

Штучний нейрон виглядає наступним чином: він на вхід отримує велику кількість сигналів, кожен з яких в той же час є виходом іншого нейрону. Далі цей вхід помножується на відповідну вагу, далі дані сумуються, визначаючи цим рівень активації нейрону.

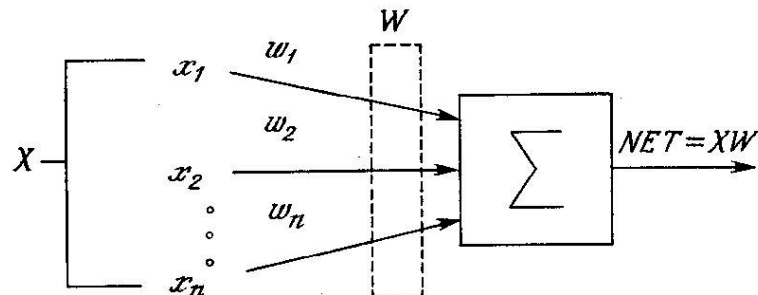


Рисунок 1.3 – Модель, яка реалізує активацію нейрону [4]

Тут на вхід нейрона надходить безліч сигналів x_1, x_2, \dots, x_n , в сукупності вони позначаються вектором X . Дані сигнали аналогічні тим, що отримує на входи біологічний нейрон. Після, сигнал множиться на відповідну вагу w_1, w_2, \dots, w_n , далі сумується в підсумовуючому блоці Σ . Кожна з ваг дорівнює силі одного синаптичного зв'язку в біологічному нейроні. Вихід, який зазвичай називається NET , створюється з підсумовуючого блоку, де зважені елементи сумуються алгебраїчно.

Сигнал NET перетворюється звичайною лінійною функцією, яка називається активаційною. Дана функція позначається F і дає вихідний сигнал OUT .

$$OUT = K(NET),$$

де K – постійна, граничної функції

$$OUT = 1, \text{ якщо } NET > T$$

$$OUT = 0 \text{ в інших випадках,}$$

T – постійна гранична величина, яка точніше моделює нейронну мережу.

На рис. 1. 4 зображено штучний нейрон с активаційною функцією:

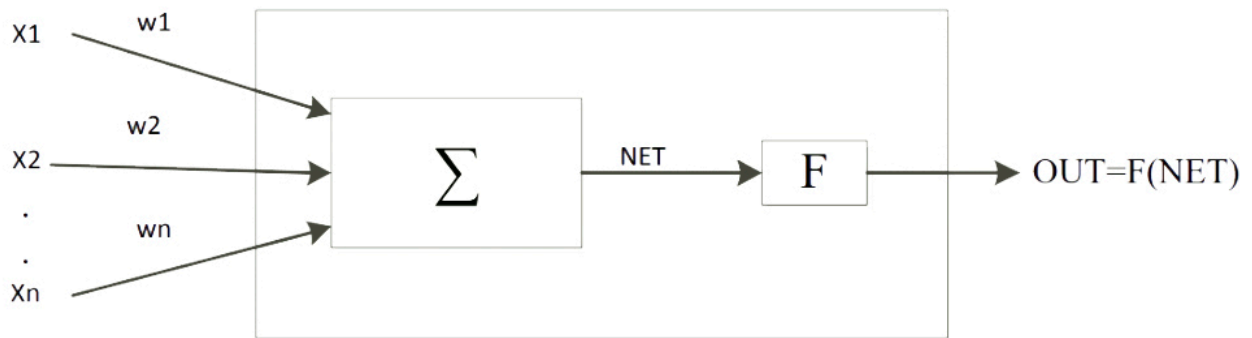


Рисунок 1.4 – Активаційна функція нейрона [7]

Блок, що позначений F , приймає сигнал NET і видає на вихід OUT .

F – являє собою функцію стискання, якщо при будь-яких значеннях NET , значення OUT належать деякому кінцевому інтервалу [8].

Модель штучного нейрона ігнорує більшість властивостей біологічного нейрона. Однією такою властивістю є затримки в часі, які впливають на динаміку всієї системи.

Вхідні сигнали відразу ж породжують вихідний сигнал. Крім того, штучний нейрон не враховує впливу синхронізуючої функції біологічного нейрона.

Та все ж, варто зазначити, що живий та штучний нейрони дуже схожі між собою.

Для того, щоб визначити місце нейронних мереж в області інформаційних технологій варто звернутися до класифікації штучних інтелектуальних систем.

1.4.2 Класифікації штучних інтелектуальних систем

Штучні інтелектуальні системи мають свої характерні особливості:

По-перше, розвинені комунікативні здібності, що характеризують спосіб взаємодії між комп'ютером, користувача з системою. Не виключена можливість звернення до системи з довільним запитом в діалозі з інтелектуальною системою. При цьому мова самої інтелектуальної системи повинна бути максимально наближеною до природної.

По-друге, вирішення погано формалізованих задач, тобто таких завдань, які не мають конкретного вирішення, та вимагають нестандартного підходу, в залежності від ситуації, існуючих даних і кінцевого результату. Погано формалізовані задачі ефективно вирішуються за допомогою штучних нейронних мереж.

По-третє, здатність до самонавчання – тобто можливості отримання знань інтелектуальною системою з накопиченого досвіду конкретних ситуацій. Для того, щоб навчити систему нам необхідні заздалегідь оброблені початкові дані. Згідно з поданими ознаками, властивостями, інтелектуальні системи можна розділити на табл. 1.1):

Таблиця 1.1 – Види штучних інтелектуальних систем

Вид штучної інтелектуальної системи	Тип штучної інтелектуальної системи
Системи з комутативними здатностями	<ul style="list-style-type: none"> – інтелектуальні бази даних; – природно-мовні інтерфейси; – гіпертекстові системи; – контекстні довідкові системи; – когнітивна графіка.
Експертні системи	<ul style="list-style-type: none"> – класифікуючі системи; – системи довизначення; – трансформуючі системи; – багатоагентні системи.
Системи автоматичного навчання	<ul style="list-style-type: none"> – індуктивні системи; – нейронні мережі; – системи основані на прецедентах; – інформаційні сховища.

Закінчення таблиці 1.1

Вид штучної інтелектуальної системи	Тип штучної інтелектуальної системи
Адаптивні системи	– CASE-технології; – компонентна технологія.

Інтелектуальні бази даних відрізняються від звичайних тим, що там є можливість зробити вибірку за запитом необхідної інформації, що може явно не зберігатися в цій базі, а виводитися з наявної в БД.

Як зрозуміло з назви, природно-мовний інтерфейс перетворює мовні конструкції природної мови у внутрішньо-машинний рівень представлення знань. Використовується для доступу до інтелектуальних баз даних, контекстного пошуку задокументованої текстової інформації, команд голосового вводу в системах управління, а також для машинного перекладу з іноземних мов.

Гіпертекстові системи застосовують у базах текстової інформації, де потрібен пошук за ключовими словами, також мають складнішу семантичну організацію ключових слів.

За допомогою контекстних довідкових систем користувач описує ситуацію або проблему, а система при цьому за допомогою додаткового діалогу старається її конкретизувати та виконує пошук рекомендацій, які найбільш відповідні даній ситуації. Дані системи створюються, як додатки для систем документації та відносяться до класу систем поширення даних.

Системи когнітивної графіки використовуються для моніторингу та управління оперативними процесами. Графічні образи описують множину параметрів досліджуваної ситуації.

Експертні системи призначені для вирішення задач в основі яких лежить накопичення великої кількості даних до БД. Ці дані відображають досвід роботи експертів в проблемній області, яка розглядається.

Багатоагентні системи являють собою динамічні системи, для яких характерна інтеграція з базою знань декількох різноманітних джерел, що обмінюються одна з одною отриманими даними на динамічній основі.

Системи автоматичного навчання – основані на методах автоматичної класифікації прикладів ситуацій з реальної практики [9, с. 12-19]. Основними види таких систем:

- системи, які навчаються "з учителем", коли для кожного прикладу у явному вигляді задається значення ознаки його приналежності до певного класу ситуацій;
- системи, що навчаються "без учителя", коли система сама визначає класи ситуацій за ступенем близькості значень ознак цієї ситуації.

Індуктивні системи, за принципом від малого до великого, узагальнюють приклади, а сам процес узагальнення відбувається наступним чином:

1. Вибирається певна ознака класифікації з безлічі заданих (послідовно або за заданим правилом).
2. За значенням вибраної ознаки, приклади розбиваються на підмножини.
3. Виконується перевірка приналежності прикладу до одного з класів.
4. Якщо певна підмножина прикладів належить одному підкласу, тобто у всіх прикладів підмножини збігається значення ознаки, що формує клас, то процес класифікації припиняється (інші ознаки класифікації, що залишилися – не розглядаються).
5. Для тих підмножин прикладів, в яких не збігаються ознаки, які формують клас – процес класифікації продовжується, починаючи з 1 пункту.

Можна сказати, що нейронна мережа – інструмент для паралельних обчислень, що складаються з безлічі простих процесорів, які періодично отримують та відправляють сигнали іншим процесорам. Далі пропоную звернутися до інтелектуальних інформаційних систем з точки зору завдань, що потрібно вирішити. Можна виділити наступні системи:

- системи комп'ютерної лінгвістики;
- системи розпізнавання образів;

- ігрові системи;
- системи створення інтелектуальних інформаційних систем [10, с.22].

На рис. 1.5 представлена класифікація інтелектуальних інформаційних систем за типом розв'язуваних задач:

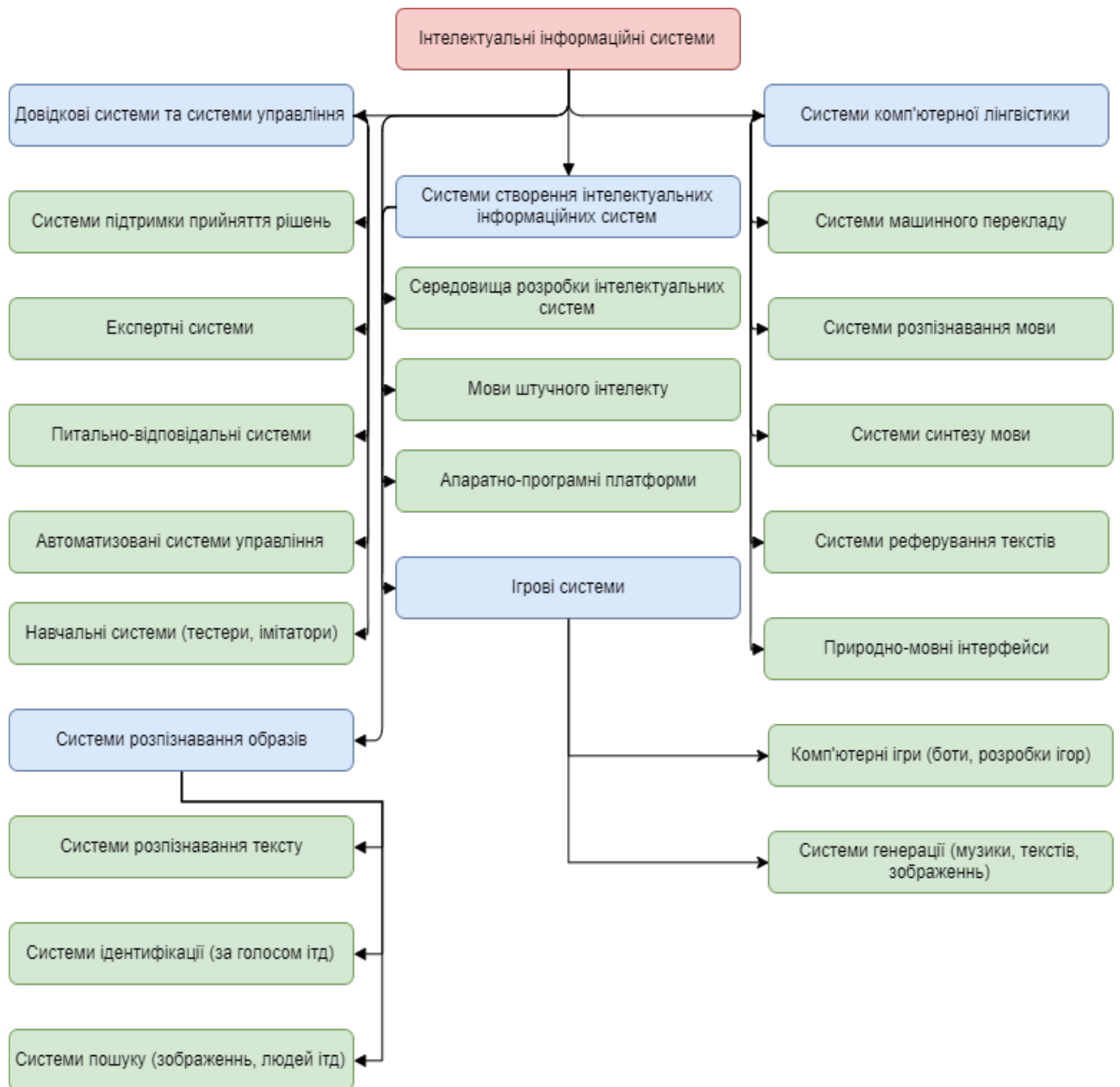


Рисунок 1.5 – Класифікація інформаційних систем за типом розв'язуваних задач

Системи можуть вирішувати не одну, а кілька задач або в процесі виконання одного завдання вирішувати і ряд інших задач. Інтелектуальні системи також

класифікуються за таким критерієм як «методи використання». Тут можна виділити м'які, жорсткі і гібридні методи, рис. 1.6:

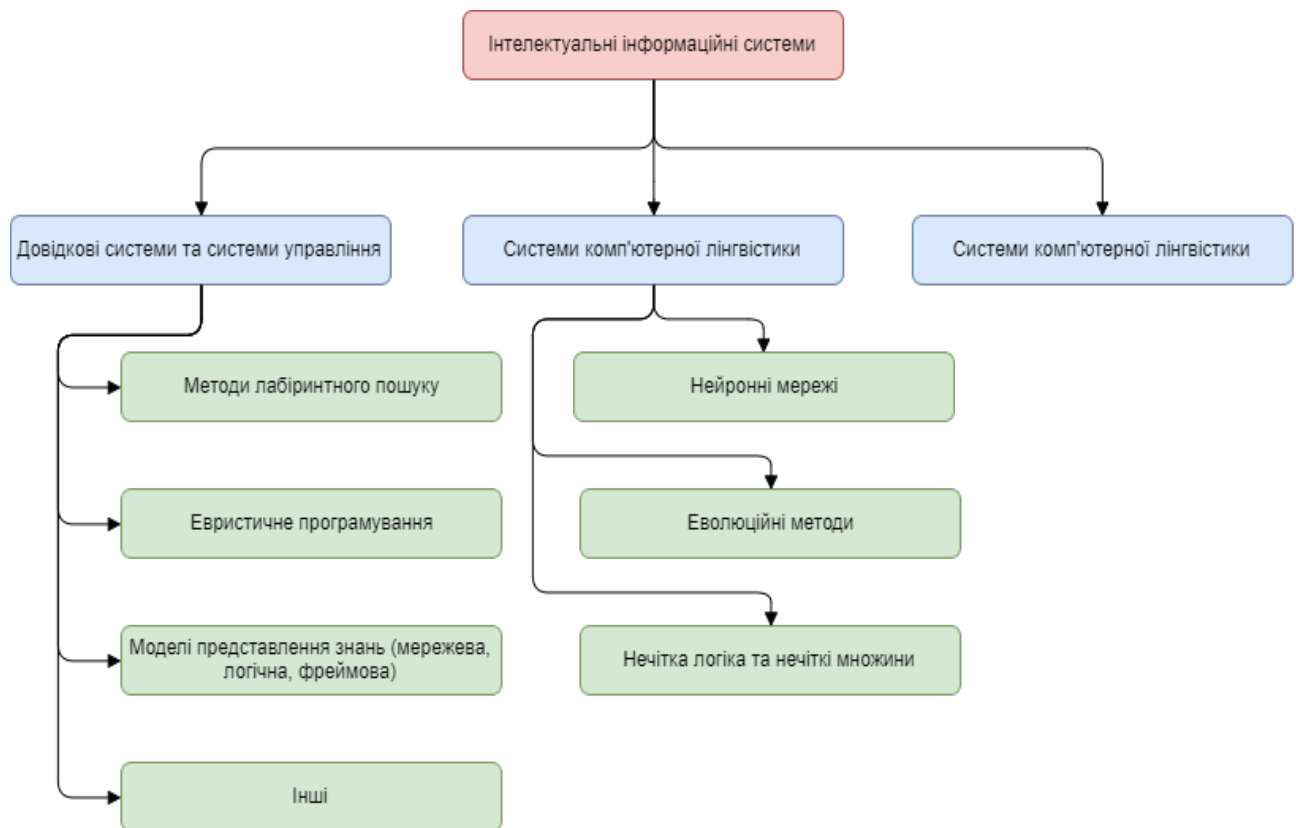


Рисунок 1.6 – Класифікація інформаційних систем за методами

М'які обчислення (Soft Computing) – це складна комп'ютерна методологія, що заснована на нечіткій логіці, генетичних обчисленнях, нейрокомп'ютензі та імовірнісних обчисленнях. Жорсткі обчислення – стандартні комп'ютерні обчислення (що не відносяться до м'яких). Гібридні системи – системи, що використовують більше, ніж одну комп'ютерну технологію (у випадку інтелектуальних систем – технології штучного інтелекту) [10, с.6]. Класифікація штучних інтелектуальних систем охоплює величезну теоретичну базу знань в різних областях науки. Створення та реалізація ІОС – складний процес від початкового до кінцевого етапів.

Тож, що таке машинне навчання, нейрон та нейронні мережі ми вже знаємо, тому пропоную перейти до пошуку та аналізу існуючих рішень.

1.5 Аналіз існуючих рішень

Для того, щоб більш детально вивчити проблему, пропоную розглянути вже існуючі системи, за допомогою яких можливо розпізнавати тварин. Провівши аналіз ринку систем розумного дому, було виявлено, що на сьогоднішній день таких на ринку немає. Також було проведено аналіз систем з розпізнавання об'єктів, а саме тварин. Таких систем існує багато, та більшість із них використовується у задачах розпізнавання предметів побуту, покупок тощо.

1.5.1 Аналіз системи «Google Lens»

Система Google Lens, яка представлена компанією Google дозволяє вирішувати задачі розпізнавання не тільки предметів, а і тварин та рослин. Дана система зазвичай використовується на мобільних пристроях для визначення предмету/об'єкту на фото – якщо це товар, система показує його ціну і пропонує перейти на сайт для його купівлі. Якщо це тварина або рослина – виводить про них інформацію, що за рід, порода тощо. З графічним інтерфейсом даної системи можна ознайомитись на рис. 1.7.

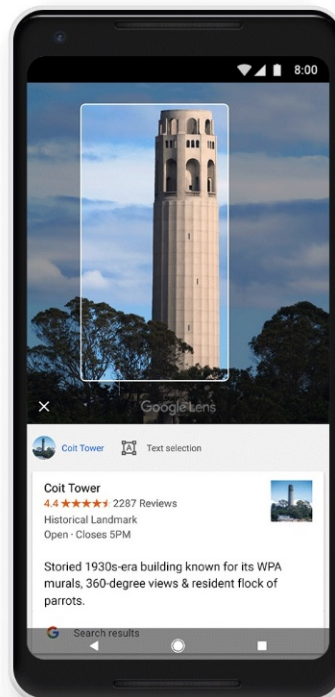


Рисунок 1.7 – Зовнішній вигляд системи «Google Lens»

Основними перевагами даної системи є:

- можливість використання на мобільних пристроях;
- швидкість визначення об'єкту;
- можливість визначення будь-якого предмету.

Серед недоліків можна виділити наступне:

- немає можливості використання даної програми у системах розумного дому;
- немає можливості навчання даної програми визначати нові об'єкти, які потрібні користувачеві;
- система працює коректно тільки у США, так як немає повної підтримки української локалізації;
- працездатність тільки на мобільних пристроях. Система не оптимізована для використання на стаціонарних комп'ютерах.

1.5.2 Аналіз системи «Image Recognizer»

Даний сервіс дозволяє визначити, який об'єкт зображено на картинці. Все, що вам потрібно зробити, щоб ідентифікувати об'єкт – сфотографувати його та відправити фото на аналіз. Далі система повинна визначити, що зображено на картинці. З візуальним інтерфейсом можна ознайомитись на рис. 1.8.

Основними перевагами даної системи є:

- можливість використання на мобільних пристроях;
- можливість визначення будь-якого предмету.

Із недоліків можна виділити:

- засіб не можливо зробити інтерактивним, тобто задати додаткові умови для визначення об'єкту на фото;
- присутня велика кількість реклами, що сильно заважає.

- працездатність тільки на мобільних пристроях. Система не оптимізована для використання на стаціонарних комп'ютерах.
- неможливо налаштувати систему для використання у системах розумного дому.

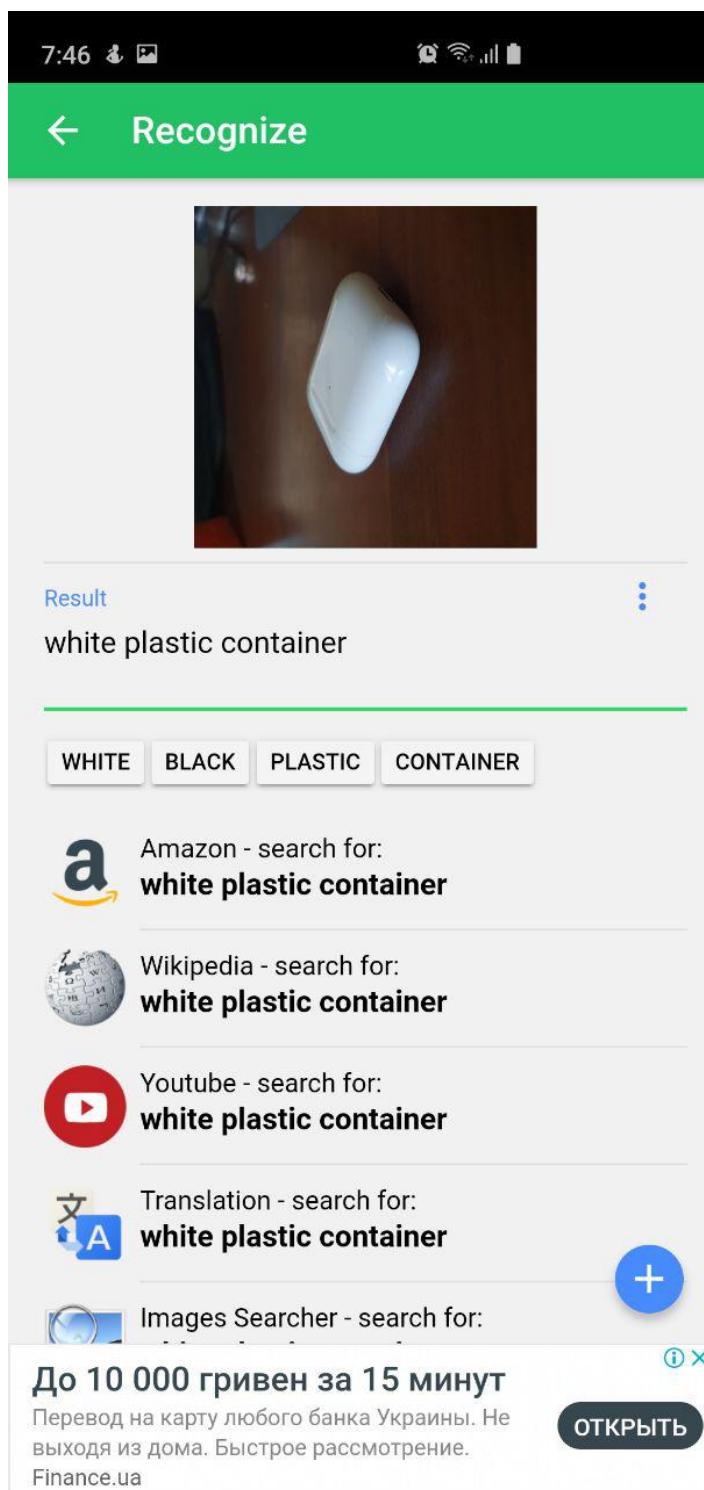



Рисунок 1.8 – Зовнішній вигляд системи «Image Recognizer»

1.5.3 Аналіз системи «Imagga»

Названий вище сервіс дозволяє не тільки визначати, а і категоризувати об'єкти, що зображені фото. Роботу програми можна побачити на рис. 1.9 та на рис. 1.10.

Auto-Tagging demo [Switch to Categorization demo](#) [Go to NSFW demo](#) [back to Imagga's homepage](#)

Upload your photo
You can upload a photo or paste a URL of an image



Note: By uploading files here you agree to have them temporarily stored in our training dataset for the sole purpose of improving Imagga's technology.

UPLOAD IMAGE

Image URL

Tip: You can paste any image URL here and get tags. ☐ Include colors (Might be slower)

Analyze




Generated tags English

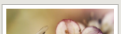

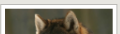
Concepts

dog	95.32%
canine	83.98%
pet	77.30%
animal	63.34%
companion	62.38%
purebred	61.25%
studio	60.92%
mammal	58.28%
domestic	57.99%
breed	53.27%

[show me more tags](#)

Try with example images
Select one of the following images to see the results:

Start using our Tagging API.
Check our [Pricing Plans](#) or:

Sign Up for Free

Implement in seconds

Shell Node.js Java Python PHP Ruby

```
querystring = {"image_url": "http://playground.imagga.com/"}
headers = {
  'accept': "application/json",
  'authorization': "Basic YWNjXzJkYzdkNzNjMmYwODU="
}
response = requests.request("GET", url, headers=headers)
print(response.text)
```


Powered by APiEmbed [Copy to Clipboard](#)

See Our Full [API Documentation](#)

Рисунок 1.9 – Зовнішній вигляд сервісу Imagga

Auto-Categorization demo [Switch to Tagging demo](#) [Go to NSFW demo](#) [back to Imagga's homepage](#)

Upload your photo
You can upload a photo or paste a URL of an image



Note: By uploading files here you agree to have them temporarily stored in our training dataset for the sole purpose of improving Imagga's technology.

UPLOAD IMAGE

Image URL




Tip: You can paste any image URL here and get categories.




Analyze

Generated categories English

Pets & Animals 100.00%

Try with example images
Select one of the following images to see the results:

Start using our Categorization API.
Check our [Pricing Plans](#) or:

Sign Up for Free

Implement in seconds

Shell Node.js Java Python Go

```
curl --request GET \
  --url 'https://api.imagga.com/v2/categories/personal' \
  --header 'accept: application/json' \
  --header 'authorization: Basic YWNjXzJkYzdkNzNjMmYwODU='
```

Powered by APiEmbed [Copy to Clipboard](#)

See Our Full [API Documentation](#)

Рисунок 1.10 – Зовнішній вигляд сервісу Imagga

До переваг даної системи можна віднести:

- дана система дозволяє не тільки визначити об'єкти на фото, а і категоризувати їх;
- можливість визначення будь-якого об'єкту на фото.

Недоліками даної системи є:

- потреба в оплаті для використання на постійній основі;
- немає можливості налаштувати дану програму для використання у системах розумного дому.

1.6 Порівняння існуючих програмних рішень

В результаті проведеного аналізу сформуємо порівняльну таблицю, на якій представлені переваги та недоліки розглянутих аналогів.

Таблиця 1.2 – Порівняння розглянутих аналогів

Критерій порівняння	Google Lens	Image Recognizer	Imagga
Функціональність	+	+	+
Зручність використання	+	-	+
Можливість розширення функціоналу	-	-	-
Кросплатформність	-	-	+
Можливість використання у системах розумного дому	-	-	-

Закінчення таблиці 1.2

Критерій порівняння	Google Lens	Image Recognizer	Imagga
Можливість безкоштовного використання	+	+	+
Наявність реклами в додатку	-	+	-
Відкрите API	-	-	+

Отже, підсумуємо результати, які отримано під час порівняння аналогів. Всі розглянуті продукти неможливо використовувати у системах розумного дому. Жоден застосунок неможливо розширити у програмному сенсі. Тільки один з аналогів являється кросплатформним, тобто його можна використовувати з будь-якого пристрою, наприклад з ПК або планшета. Всі системи можна використовувати безкоштовно, але для того, щоб мати розширений функціонал або прибрати рекламу, яка дратує, потрібно заплатити.

Основними критеріями для майбутнього сервісу є:

- Функціональність;
- Зручність у використанні;
- Додаток не міститиме дратуючої реклами;
- Можливість використання у системах розумного дому;
- Кросплатформність;

На старті проекту, сервіс матиме підтримку тільки англійської локалізації. В майбутньому планується додати нові локалізації.

1.7 Постановка задачі

Метою даної роботи є удосконалення систем розумного дому, а саме створення додаткового модуля, який дозволить визначати домашнього улюбленця на фото (кадрі із відео) та допускати його до будівлі. При цьому,

додатково, користувач отримуватиме сповіщення про те, що домашня тварина прийшла під двері дому. За допомогою даного сповіщення користувач знатиме, що його домашня тварина повернулася додому, та за допомогою додатку управління системою розумного дому, зможе відчинити спеціальний отвір, через який тварина матиме можливість увійти до будівлі.

Для досягнення даної мети потрібно вирішити наступні завдання:

- провести аналіз систем, які використовують машинне навчання та нейронні мережі. Що таке машинне навчання та нейронні мережі ми вже знаємо – це описано в даному розділі;
- провести додатковий аналіз видів нейромереж;
- визначити, які види навчання нейромереж існують;
- розробити веб-застосунок, котрий реалізуватиме функціонал розпізнавання домашніх тварин;
- розробити функціонал відправлення сповіщень користувачеві даної системи;
- розробити зручний інтерфейс для користувачів;
- навчити нейронну мережу визначати тварин на фото.

Розроблений веб-застосунок повинен відповідати загальноприйнятим стандартам та визначеним вище вимогам, а саме:

- забезпечення швидкого доступу до системи;
- забезпечення швидкого доступу до даних;
- висока стійкість до відмов;
- система повинна бути простою у використанні;
- система повинна легко масштабуватись;
- система повинна легко налаштовуватися.

Отже, вимоги до застосунку визначено, можемо переходити до вибору засобів, які потрібні для реалізації.

Висновки до розділу

В даному розділі описано об'єкт та предмет дослідження, оглянуто та проаналізовано деякі існуючі рішення, виділено основні інкапсульовані завдання, котрі потрібно вирішити для побудови визначеної системи. Також сформовано вимоги до системи, за якими буде вестися розробка, і вони ж будуть слугувати критеріями для перевірки на відповідність отриманого результату до поставлених цілей.

РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Типи нейронних мереж

2.1.1 Одношарові штучні нейронні мережі.

Зі слів "нейронна мережа", можна зрозуміти, що це така найпростіша мережа[11], яка складається з групи нейронів, які утворюють шар (layer), як показано в правій частині на рис. 2.1.

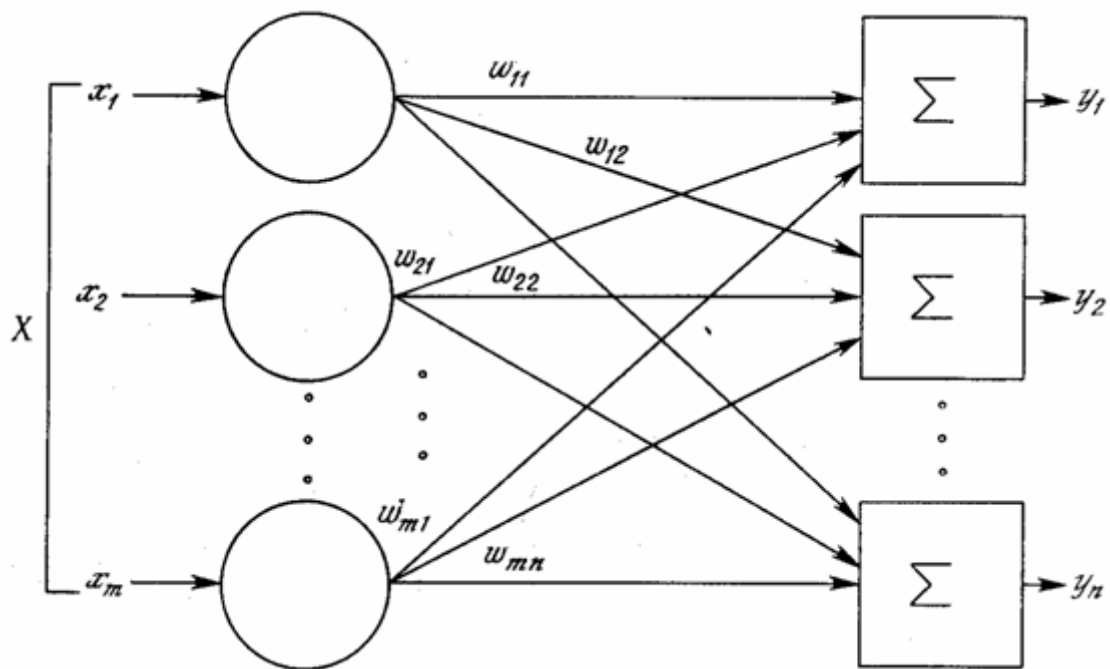


Рисунок 2.1 – Одношарова нейронна мережа

Мушу зазначити, що кола-вершини, що знаходять зліва на зображенні слугують тільки лиш для розподілу вхідних сигналів і тільки для цього. Ці кола не виконують жодних обчислень, тому і не вважають шаром. Саме тому їх позначено колами, щоб відрізнити їх від нейронів, які роблять певні обчислення, вони позначені квадратами. Всі елементи з множини входів X сполучені окремою вагою зі всіма штучними нейронами. І кожен нейрон віддає зважену суму входів в мережу. У біологічних та штучних нейронів багато з'єднань можуть бути відсутніми, всі з'єднання показано з метою спільності. Також, можуть мати місце з'єднання між виходами і входами елементів в шарі[12].

W – ваги всіх елементів матриці. Дана матриця має m рядків та n стовпців, де m – кількість входів, n – кількість нейронів.

Для прикладу, $w_{2,3}$ – являє собою вагу, яка зв'язує третій вхід з другим нейроном. Таким чином, обчислення вихідного вектора N , компонентами якого є виходи OUT нейронів, зводиться множення матриць $N = XW$, де N та X – вектори-рядки.

2.1.2 Багат шарові штучні нейронні мережі.

Більш складні та більші мережі, як правило володіють великими обчислювальними можливостями. На сьогоднішній день створено багато мереж, будь-яких конфігурацій, багат шарова організація нейронів копіює структури певних відділів мозку людини. Виявляється, що такі багат шарові мережі володіють більшими можливостями, ніж одношарові, та останнім часом розроблено багато алгоритмів для їх навчання. Багат шарові мережі можуть будуватися з шарів, які складають каскади. Виходом одного шару може бути вхід для наступного шару. Схожа мережа зображена на рис. 2.2, разом зі всіма сполученнями.

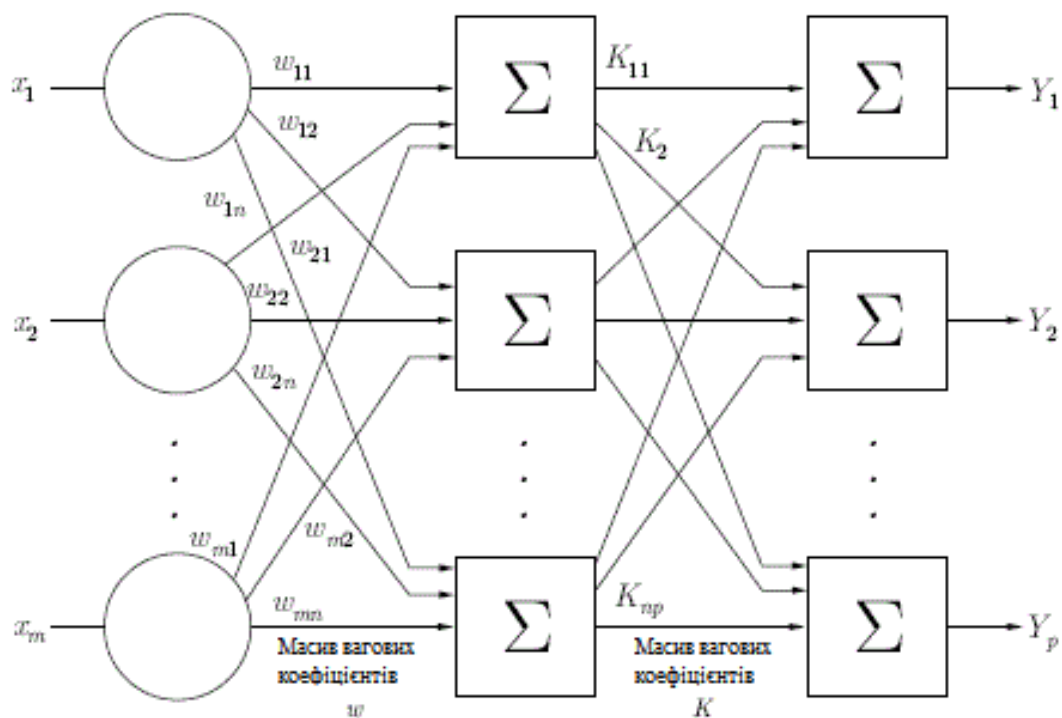


Рисунок 2.2 – Двошарова нейронна мережа

Якщо активаційна функція між шарами є лінійною, то не факт, що багатошарова мережа виявиться ефективнішою за одношарову. Обрахунок виходу шару полягає в множенні вхідного вектора на першу матрицю ваги з подальшим множенням (якщо нелінійна активаційна функція відсутня) результуючого вектора на другу матрицю ваги:

$$OUT = (XW_1)W_2.$$

Оскільки множення матриць асоціативне, то:

$$(XW_1)W_2 = X(W_1W_2).$$

Звідси видно, що двошарова лінійна мережа еквівалентна одному шару з ваговою матрицею, яка дорівнює добутку двох вагових матриць[13]. Отже, будь-яку багатошарову лінійну мережу можна замінити на еквівалентну одношарову. Однак, одношарові мережі достатньо обмежені за своїми обчислювальними можливостями. Тож, для того щоб розширити можливості мереж в порівнянні з одношаровою мережею нам необхідна нелінійна активаційна функція.

2.1.3 Рекурентні нейронні мережі

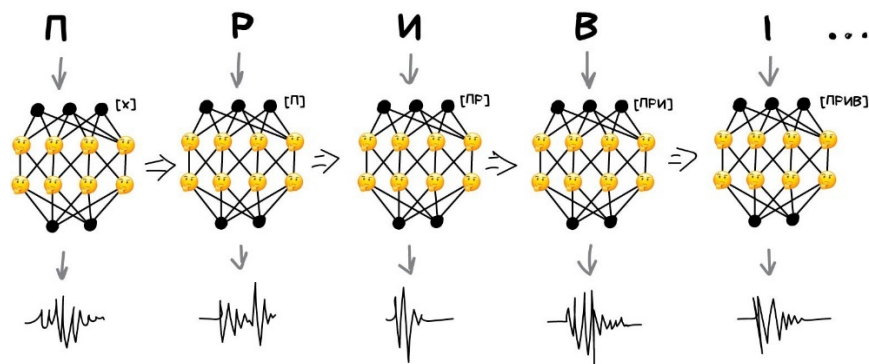
Дана архітектура дуже популярна та займає друге місце за частотою використання. Дякуючи рекурентним мережам ми можемо користуватися можливостями комп'ютерного синтезу та машинного перекладу текстів. Всі завдання, що хоч якось пов'язані з послідовностями, розв'язуються даними мережами. Це можуть бути такі послідовності: текстові, музичні, голосові тощо.

Згадайте колишні синтезатори речі, такі як Microsoft Sam. Дані синтезатори мали на меті вимовляти слова, як це робимо ми – люди. Але все, що у них виходило, це вимова слів по буквах, що звучало дуже смішно. Та системи, які існують зараз вміють не тільки вимовляти слова без помилок, а ще й розставляти правильно акценти в реченнях. Ось назви таких систем: "Google Assistant" компанії Google, "Аліса", розробником якої являється компанія Яндекс. Таких систем на даний момент дуже багато.

Ми не можемо змусити неймережу одразу видавати нам фрази повністю, бо їй прийдеться запам'ятати всі мовні фрази, що займе неймовірно велику кількість пам'яті. Завдяки тому, що мова, текст чи музика – послідовність, це дозволяє вирішити проблему. Кожен звук або слово є самостійною одиницею, яка є залежною від попередніх. Коли такий зв'язок втрачено – виходить щось нерозбірливе.

Мережу досить легко навчити вимовляти окремі букви, чи навіть слова. Нам знадобиться велика кількість розмічених на слова аудіо файлів. Далі ми навчаємо нашу мережу за вхідним словом видавати послідовність сигналів, схожих на людську вимову. Далі порівнюємо цю послідовність з оригіналом та намагаємося досягти максимального результату. Для таких цілей підійде перцептрон.

Оскільки перцептрон не вміє запам'ятовувати того, що він генерував раніше, у нас знову з'являється проблема з послідовністю. Кожен раз, коли перцептрон завершує роботу, він "забуває" попередні обчислення. Після, з'явилася ідея кожен нейрон наділити пам'яттю. Так придумали рекурентні мережі, де всі нейрони пам'ятають попередні відповіді, і з кожним новим запуском використовують їх як додаткові входи. Таким чином, нейрон може в майбутньому сам собі дати підказку, що та чи інша відповідь правильна, а інша – ні. Наприклад, що звук, який йде далі, має звучати на пів-тона вище, тому що буква в послідовності є голосною.



Рекурентна Неймережа (RNN)

Рисунок 2.3 – Рекурентна нейронна мережа

2.1.4 Нейронні мережі прямого поширення та перцептрон

Нейронні мережі прямого поширення (рис. 2.4) та перцептрони (рис. 2.5) (perceptrons, P) є дуже прості – вони передають інформацію від входу до виходу. Вважається, що у нейронних мереж є шари, кожен з яких складається з вхідних, прихованих або вихідних нейронів. Нейрони одного шару між собою не пов'язані, при цьому кожен нейрон цього шару пов'язаний з кожним нейроном сусіднього шару[13].

Найпростіша робоча мережа складається з двох вхідних і одного вихідного нейрона і може моделювати логічний вентиль – базовий елемент цифрової схеми, що виконує елементарну логічну операцію. FFNN зазвичай навчають методом зворотного поширення помилки, подаючи моделі на вхід пари вхідних і очікуваних вихідних даних.

Під помилкою зазвичай розуміються різні ступені відхилення вихідних даних від вихідних (наприклад, середньоквадратичне відхилення або сума модулів різниці). За умови, що мережа має достатню кількість прихованих нейронів, теоретично вона завжди зможе встановити зв'язок між вхідними та вихідними даними. На практиці використання мереж прямого поширення обмежена, і частіше вони використовуються спільно з іншими мережами.

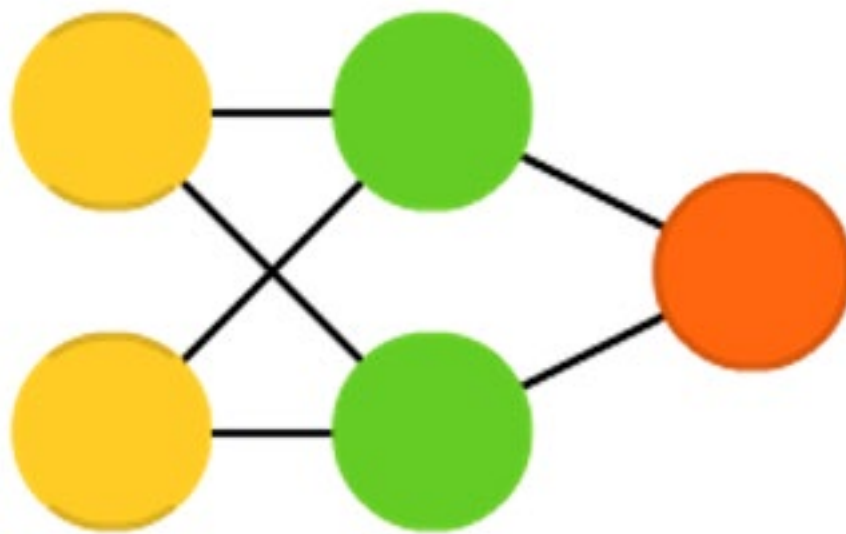


Рисунок 2.4 – Нейронна мережа прямого поширення

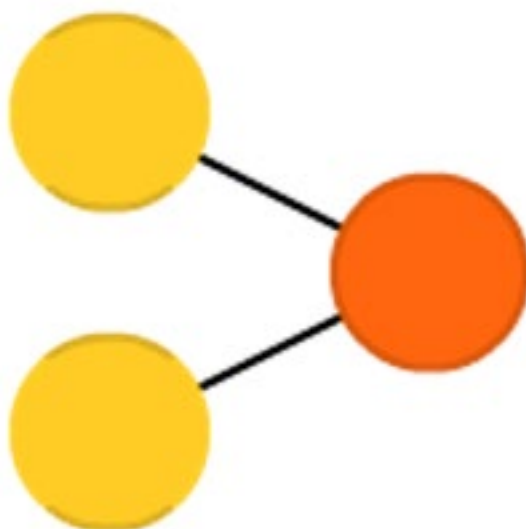


Рисунок 2.5 – Перцептрон

2.1.5 Нейронна мережа Гопфілда

Нейронна мережа Гопфілда (рис. 2.6) – повнозв'язна мережа (кожен нейрон з'єднаний один з одним), де кожен нейрон виступає у всіх трьох іпостасях. Кожен нейрон є вхідним до навчання, прихованим під час нього і вихідним після.

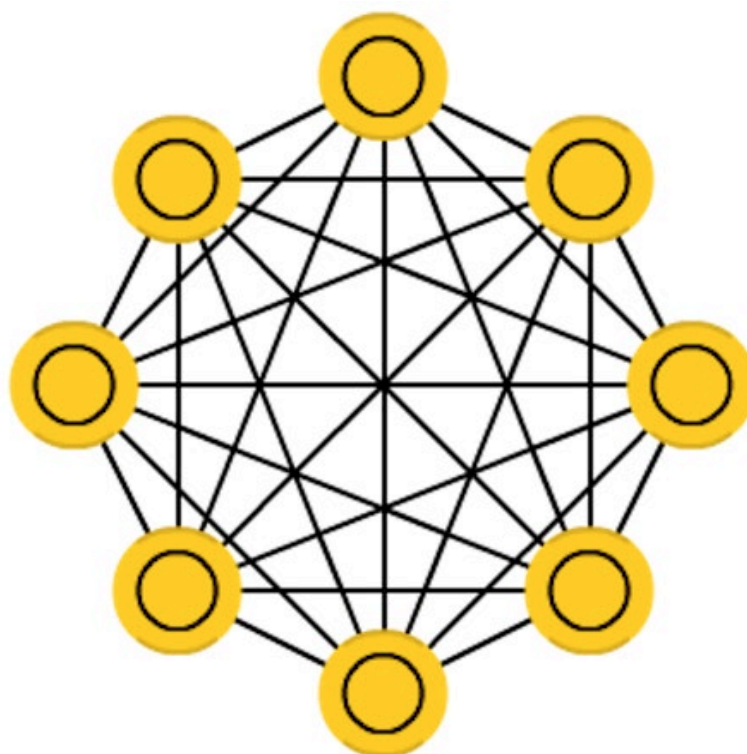


Рисунок 2.6 – Вигляд нейронної мережі Гопфілда

Матриця ваг підбирається таким чином, щоб всі «успішні реєстрації» вектора були б для неї власними. Одного разу навчена на одному або декільком образах система буде сходиться до одного з відомих їй образів, тому що тільки один із цих станів є стаціонарним[14].

Варто відзначити, що це не завжди відповідає очікуваному стану. Система стабілізується тільки частково, тому що загальна "енергія" мережі під час навчання поступово знижується. У кожного нейрона є поріг активації, співрозмірний з цією енергією, і якщо сума вхідних даних перевищить цю межу, нейрон може переходити в один з двох станів (зазвичай -1 або 1, інколи 0 чи 1). Вузли мережі можуть оновлюватися паралельно, але зазвичай це відбувається послідовно. В останньому випадку створюється випадкова послідовність, що визначає порядок, в якому нейрони будуть оновлювати свій стан. Коли кожен з нейронів оновився і їх стан більше незмінний, мережа повертається до стаціонарного стану.

Такі мережі часто називають асоціативною пам'яттю, бо вони сходяться зі станом, що є найбільш близьким заданому: як і людина, коли бачить тільки половину картинки, може домалювати частину, якої не вистачає – так і нейронна мережа, отримуючи на вхід наполовину зашумлену картинку, добудовує її до цілої.

2.1.6 Ланцюги Маркова

Ланцюги Маркова (рис. 2.7) – свого роду попередники машин Больцмана (BM) і мереж Гопфілда (HN). У ланцюгах Маркова ми задаємо ймовірності переходу з поточного стану в сусідні. Крім того, дані ланцюги не мають пам'яті: подальший стан залежить тільки від поточного і не залежить від всіх минулих станів[15]. Хоча ланцюг Маркова не можна назвати нейронною мережею, він близький до них і формує теоретичну основу для BM і HN. Ланцюги Маркова також не завжди є повнозв'язними.

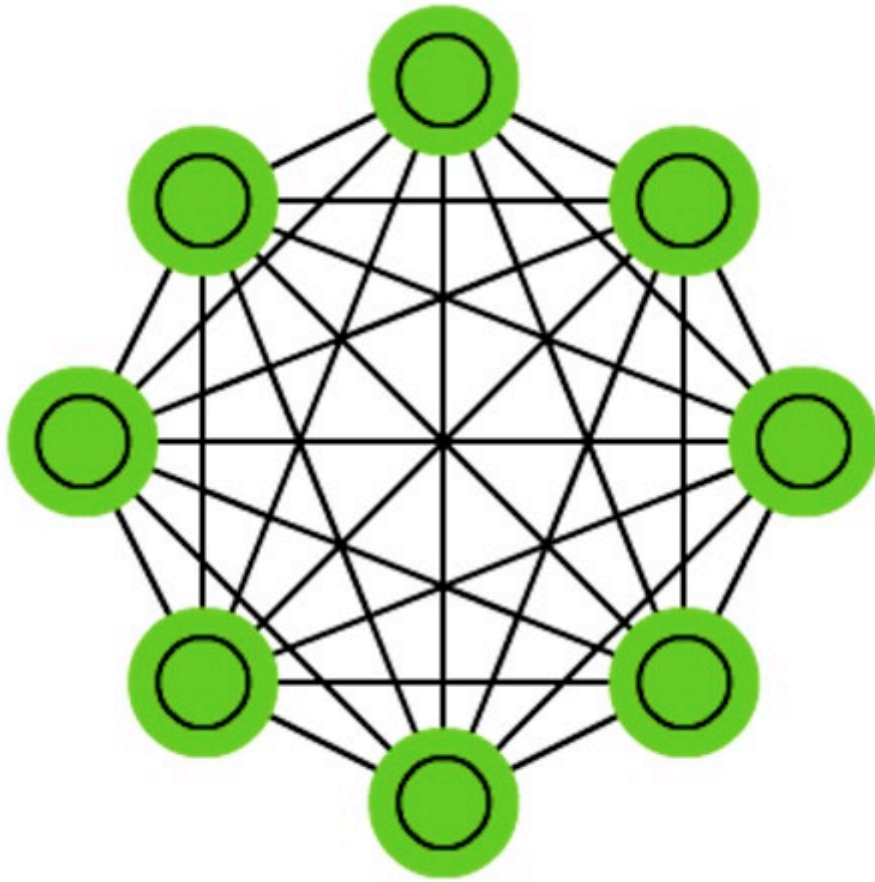


Рисунок 2.7 – Ланцюги Маркова

2.1.7 Машина Больцмана

Машини Больцмана (рис. 2.8) багато в чому схожі на мережі Гопфілда, але в них деякі нейрони позначені як вхідні, а деякі залишаються прихованими[16]. Вхідні нейрони стають вихідними, коли всі нейрони в мережі оновлюють свої стани. Спочатку вагові коефіцієнти присвоюються випадковим чином, потім відбувається навчання методом зворотного поширення. ВМ – стохастична нейронна мережа, так як в навчанні задіяно ланцюг Маркова. Процес навчання та роботи тут майже такий же, як і в мережі Гопфілда: нейронам задають певні початкові стани, а потім ланцюг починає вільно функціонувати. В процесі роботи нейрони можуть приймати будь-який стан, і ми постійно переміщаємося між вхідними і прихованими нейронами. Активация регулюється значенням загальної температури, при зниженні якої скорочується і енергія нейронів.

Скорочення енергії викликає стабілізацію нейронів. Таким чином, якщо температура задана вірно, система досягає рівноваги.

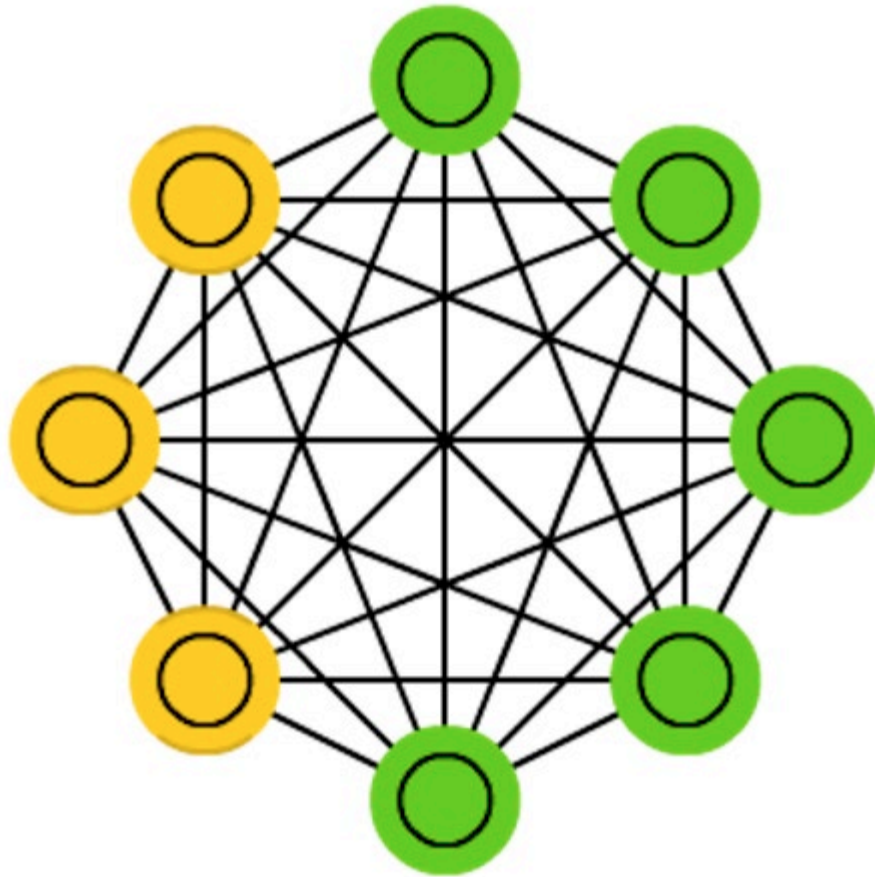


Рисунок 2.8 – Машина Больцмана

2.1.8 Згорткові нейронні мережі

Згорткові нейронні мережі та глибокі згорткові нейронні мережі кардинально відрізняються від інших мереж. Вони використовуються в основному для обробки зображень, іноді для аудіо та інших видів вхідних даних[17]. Типовим способом застосування CNN є класифікація зображень: якщо на вхід подається зображення кішки, мережа видасть «кішка», якщо картинка собаки – «собака». Такі мережі зазвичай використовують «сканер», який не обробляє всі дані за один раз. Наприклад, якщо у вас є зображення 200x200, і ви хочете будувати шар мережі з 40 тисяч вузлів. Замість цього мережа обраховує квадрат розміром 20x20 (зазвичай починає з лівого верхнього кута),

потім зрушить на 1 піксель і обрахує новий квадрат, і т.д. Зауважте, що ми не розбиваємо зображення на квадрати, а скоріше повземо по ньому. Ці вхідні дані потім передаються через згорткові шари, в яких не всі вузли з'єднані між собою. Замість цього кожен вузол з'єднаний тільки зі своїми найближчими сусідами. Ці шари мають властивість стискуватися з глибиною, причому зазвичай вони зменшуються на який-небудь з дільників кількості вхідних даних (наприклад, 20 вузлів в наступному шарі перетворюються в 10, в наступному – в 5), часто використовуються степені двійки. Крім згортальних шарів є також так звані шари об'єднання (pooling layers). Об'єднання – це спосіб зменшити розмірність одержуваних даних, наприклад, з квадрата 2x2 вибирається і передається найбільш червоний піксель. На практиці до кінця CNN прикріплюють автоенкодери для подальшої обробки даних. Такі мережі називаються глибокими.

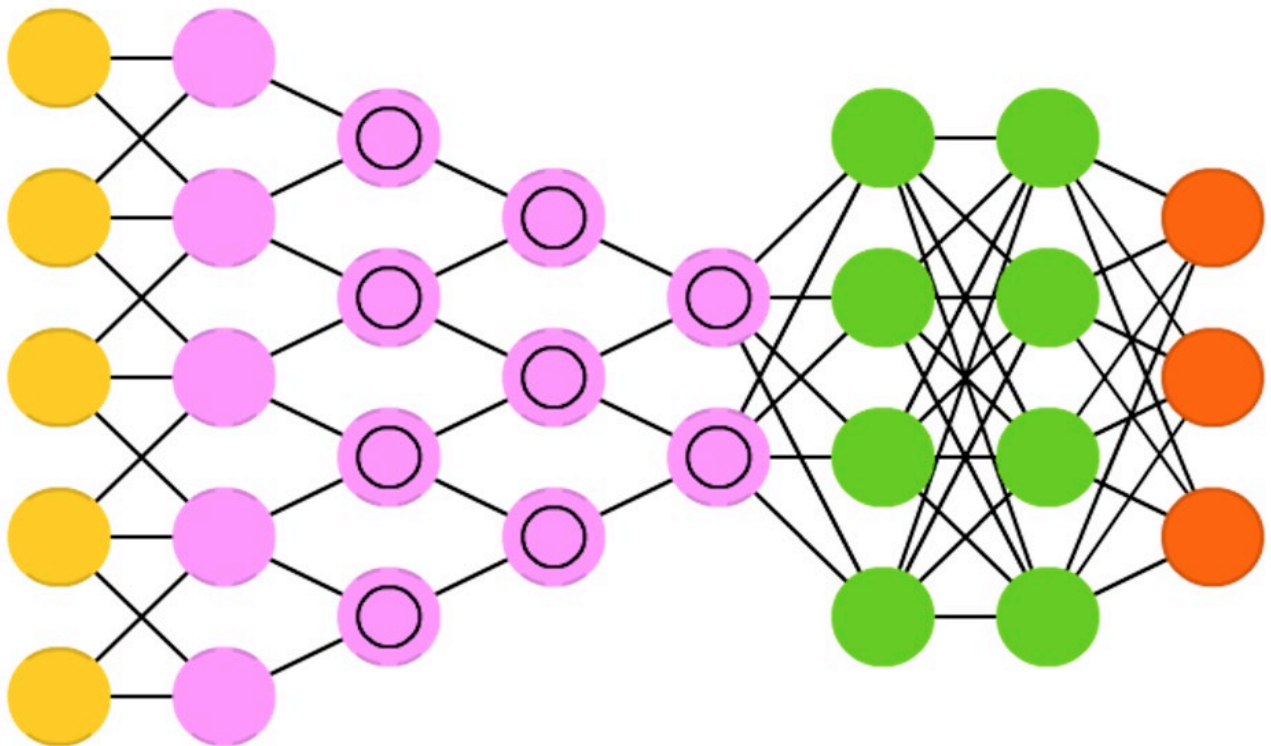


Рисунок 2.9 – Згорткова нейронна мережа

2.2 Вибір нейронної мережі для розпізнавання тварин на фото

Після проведеного аналізу типів нейронних мереж, для системи, яка повинна розпізнавати тварин на фото, потрібно використовувати згорткову нейронну мережу – систему, яка спеціально створена для визначення об'єктів на фото або зображеннях.

2.2.1 Алгоритм роботи згорткової нейромережі

Згорткова мережа має таку назву, тому що зазвичай використовує операцію під назвою згортка для оброблення зображень. Мережа використовує три види шарів:

- згортковий шар;
- шар субдискретизації;
- повнозв'язний шар.

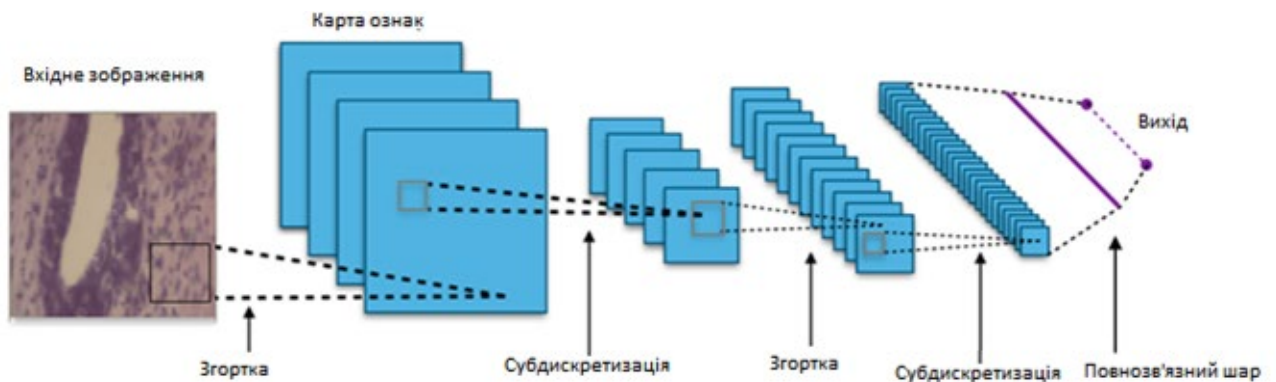


Рисунок 2.10 – Структура згорткової нейромережі

Алгоритм роботи:

1. Нам знадобиться зображення розмірністю 32x32 пікселі.
2. Вхідні дані [32x32x3] містять у собі вихідну інформацію про дане зображення (в нашому випадку 32 – це ширина, 32 – висота, та 3 – R, G, B канали).

3. Шар згортки перемножує вихідні значення пікселів на значення фільтра, далі отримані добутки сумуються (рис. 2.11). Цей процес і називається згорткою.

1	1	1	0	0
0	1	1	1	0
0	0	1x1	1x0	1x1
0	0	1x0	1x1	0x0
0	1	1x1	0x0	0x1

4	3	4
2	4	3
2	3	4

Рисунок 2.11 – Згортка зображення фільтром

4. Далі шар пулінгу виконує операцію зменшення розмірності, в кінцевому результаті обсяг може сягати $[16 \times 16 \times 12]$. Принцип роботи такий: якщо на попередньому кроці операції згортки було виявлено деякі ознаки, то образ “ущільнюється” до менш детальної картинки, та для подальшої обробки такий детальний образ вже не потрібен.
5. Завершальним етапом шар, який має назву повнозв’язного виводить N-мірний вектор для визначення класу. Даний процес відбувається зверненням до виходу попереднього шару, та визначення властивостей, що є найбільш характерними для певного, невизначеного класу.

2.2.2 Основні види навчання згорткових нейромереж

Нейронна мережа сама по собі нічого не варта. Для того, щоб вона видавала результат – її потрібно цьому навчити. Що являє собою процес навчання? Це такий процес, під час якого параметри цієї мережі налаштовуються

за деяким алгоритмом (даний алгоритм має назву алгоритму навчання) таким чином, щоб дана мережа на виході видавала максимально вірні результати. Основними алгоритмами навчання є алгоритми з учителем та без учителя.

Суть навчання з учителем заключається в тому, що у нас є готовий набір даних, де всі елементи розділені на класи, тобто система знає правильні відповіді та підлаштовує свої параметри для видачі правильного результату[20]. Наприклад, у нас є два класи тварин, собаки та коти. На вхід мережі поступає зображення тварини, яка підпадає під один із класів, і після обрахувань така мережа видає нам на виході інформацію про те, до якого з класів належить тварина, що зображена на фото. Так як, для окремого вхідного зображення нейромережа знає правильну відповідь, за допомогою функції втрат ми можемо обрахувати помилку, а саме різницю між виходом та правильним результатом, і далі відкоригувати коефіцієнти ваг. Ми продовжуємо повторювати процедуру навчання до тих пір, поки помилка не буде в межах допустимої норми. Схему того, як навчити нейронну мережу можна побачити на Додатку 1.

При навчанні без учителя мережа не знає правильних відповідей. Вона вчиться самотужки за певними ознаками, визначати клас об'єкта. Даний алгоритм сильно програє алгоритму навчання з учителем, самі розумієте чому.

Висновки до розділу

Машинне навчання дозволяє розв'язувати задачі, які важко розв'язати за допомогою конкретного алгоритму, де необхідно постійно адаптуватись до нових, непередбачених входів. До такого класу задач належить і задача розпізнавання – віднесення об'єкту до певного відомого класу. Саме розпізнавання поділяється на кілька підзадач:

- виділення об'єкту,
- виділення ознак на ньому,
- віднесення до певного класу.

В якості класифікатора, що відносить виділений об'єкт до певного класу, може використовуватись нейронна мережа. Вона представляє собою математичну модель біологічних нейронних систем, що складаються з нейронів та зв'язків між ними. Ідея цієї моделі полягає в тому, що після проведення навчання (налаштування ваг нейронів), така модель зможе працювати з новими, раніше невідомими, даними. Для розпізнавання зображень найчастіше використовують згорткову нейронну мережу, оскільки її головною особливістю є тривимірна розмірність нейронів, що зменшує загальну кількість зв'язків між усіма нейронами (не всі шари мають бути повнозв'язними). Завдяки цьому ця мережа краще працює саме з такими ієрархічними даними, як зображення (тобто стає можливим виділення додаткових ознак з уже виділених ознак). Обрано згорткову мережу для використання її у застосунку для визначення домашніх тварин на фото. Визначено, як можна навчити нейронну мережу. Для навчання буде задіяно навчання “з учителем”.

РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір архітектури та платформи розробки

Для того, щоб реалізувати поставлені задачі було обрано трирівневу архітектуру застосування, вона складається з таких частин:

- частина для зберігання даних (сервер баз даних);
- серверна частина;
- частина, яку видно клієнтові додатку (клієнтська частина).

Щоб реалізувати спроектовану архітектуру було обрано наступні технології:

- платформа .NET Core 3.0;
- C#, як мова програмування;
- MS SQL Server 2019, для того зберігання даних до БД;
- ML.NET – бібліотека машинного навчання;
- Entity Framework Core – технологія для доступу до даних, що знаходяться в БД;

3.1.1 Трирівнева архітектура

Обрана архітектура основана на середовищі клієнт-сервер. Системи, які побудовані на основі такої архітектури зазвичай розділені на три шари:

1. Шар, який бачать кінцеві користувачі.

Цей шар являє собою графічну оболонку. Зазвичай систему проектують так, щоб даний шар відповідав тільки за відображення даних. Даний шар не повинен містити в собі операцій, типу обрахунків чи зв'язків з БД.

Зазвичай даний шар виконує такі найпростіші дії:

- відображення даних, що надійшли від серверу;
- валідація оновлених даних для зберігання у БД;
- відправка оновлених даних, які пройшли валідацію на сервер;

2. Серверний шар або шар обробки даних.

Як зрозуміло із назви, даний шар відповідає за оброблення даних. Цей шар являє собою сервер. В даному шарі розміщено всю бізнес-логіку застосунку.

3. Шар доступу до збережених даних (до БД)

Цей рівень містить у собі сервер баз даних, що забезпечує зберігання нових або відредагованих даних.

За рахунок того, що система розділена на три частини, ми не маємо залежності від механізмів зберігання даних. Це дозволяє змінювати додаток так, що кінцевий користувач навіть не знатиме, що відбулися якісь зміни.

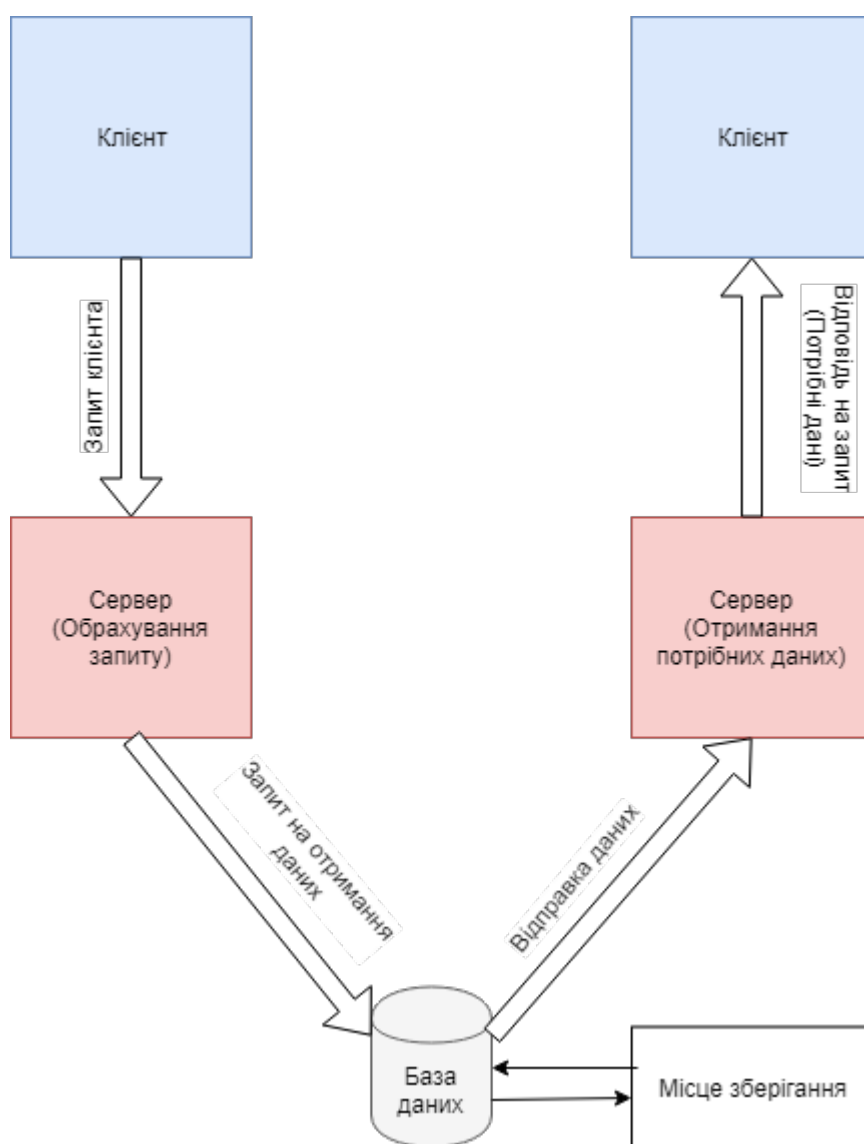


Рисунок 3.1 – Схематичне зображення трирівневого застосунку

3.1.2 Платформа Microsoft .NET Core 3.0

.NET Core 3.0[19] – модульна платформа для створення веб-застосунків, виробництва компанії Microsoft. Дана платформа є кросплатформною і з відкритим кодом. Сумісна з такими операційними системами, як Linux, Windows та macOS. Має підтримку таких мов програмування: Visual Basic . Net (тільки частково), F# та C#. В основі цієї платформи лежить платформа .NET Framework, теж від Microsoft. .NET Core відрізняється від неї модульністю, та можливістю використання хмарних технологій.

ASP.NET Core Architecture

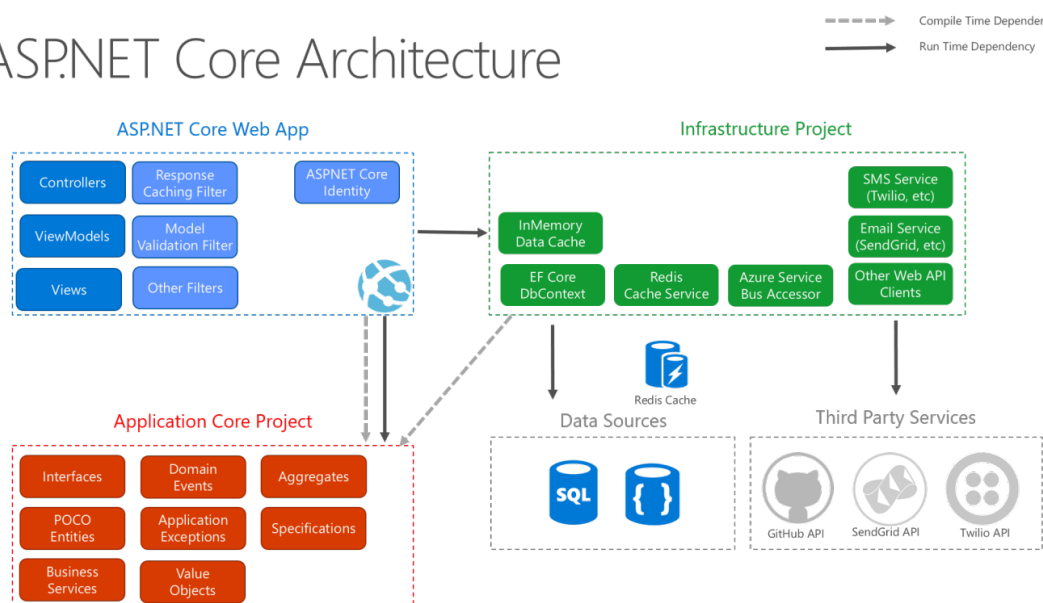


Рисунок 3.2 – Схема архітектури ASP.NET Core

ASP.NET Core Architecture

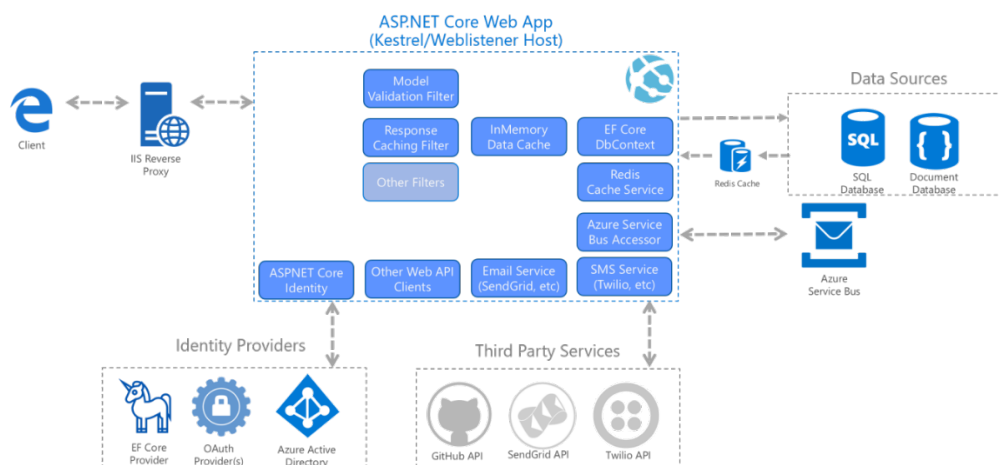


Рисунок 3.3 – Приклад архітектури застосунку під час виконання

Оскільки ця платформа є модульною, нам не потрібно оновляти її повністю, як це було із .NET Framework. Для оновлення всього лиш потрібно завантажити потрібний модуль з репозиторію за допомогою менеджера пакетів NuGet. Кожен застосунок, що написано на .NET Core, може працювати з різними модулями, які потрібні саме в даний момент. Ці модулі не залежні від єдиного оновлення платформи.

3.1.3 Мова програмування C#

C# – є сучасною та однією з найпотужніших мов програмування, є об'єктно-орієнтованою та дуже гнучкою. Це основна мова, яка використовується у .NET Framework, та .NET Core. Використовуючи цю мову програмування, можна створювати будь-які проекти, такими можуть бути:

- високонавантажені веб-портали;
- мобільні додатки;
- десктопні застосунки на будь-який смак;
- додатки, які використовують штучні нейронні мережі;
- ігри.

Дана мова забезпечує розробників великими можливостями та гнучкість при створенні програмного забезпечення. Також, вона гарантує, що додаток який написано цією мовою працюватиме довгий період часу. Також вона часто оновлюється, при цьому виправляються помилки та додається новий функціонал, що полегшує роботу для кінцевих користувачів – розробників.

3.1.4 СУБД Microsoft SQL Server 2019

MS SQL Server 2019 – являється дуже потужним інструментом для управління базами даних. Дана система надає засоби для створення та управління реляційними базами даних. Будь-який додаток, що написано на

платформі .NET Core мають змогу отримати доступ до даних, які знаходяться в даній СУБД.

Перевагами даної СУБД є:

- масштабованість;
- відмовостійкість;
- робота з хмарою;
- аналіз даних на ходу;
- висока степінь безпеки;
- висока продуктивність.

Реляційну модель баз даних у наш час використовують дуже часто. Роблять це а рахунок того, що:

А) є повна незалежність даних – для зміни структури моделі, потрібно прикласти мінімальні зусилля;

Б) дана модель являється простою – дані відображаються у зрозумілій та дуже зручній формі, бо основною структурної одиницею являється таблиця.

В) користувач, без жодних зусиль може отримати базу дану, з потрібними йому характеристиками, це досягається за допомогою методів нормалізації відношень.

3.1.5 Технологія Entity Framework Core

Entity Framework Core (EF Core) являє собою об'єктно-орієнтовану та легко масштабовану технологію від компанії Microsoft, яка використовується для доступу до даних. EF Core є ORM-інструментом (object-relational mapping – відображення даних на реальні об'єкти). Тобто EF Core дозволяє працювати з базами даних, але має і більш високий рівень абстракції: EF Core дозволяє абстрагуватися від самої бази даних і її таблиць і працювати з даними незалежно від типу сховища. Якщо на фізичному рівні ми оперуємо таблицями, індексами, первинними і зовнішніми ключами, але на концептуальному рівні, який нам пропонує Entity Framework, ми вже працюємо з об'єктами.

Entity Framework Core підтримує безліч різних систем баз даних. Таким чином, ми можемо через EF Core працювати з будь-якою СУБД, якщо для неї існує потрібний провайдер.

Microsoft підтримує ряд вбудованих провайдерів: для роботи з SQLite, MS SQL Server, PostgreSQL. Також є провайдери від сторонніх постачальників, наприклад, для MySQL.

EF Core надає універсальний API для роботи з даними. І якщо, наприклад, ми вирішимо змінити цільову СУБД, то основні зміни в проекті будуть стосуватися насамперед конфігурації і настройки підключення до відповідних провайдерів. А код, який безпосередньо працює з даними, отримує дані, додає їх в БД і т.д., залишиться колишнім.

Entity Framework Core багато успадкував від своїх попередників, зокрема, Entity Framework 6. У той же час треба розуміти, що EF Core – це не нова версія по відношенню до EF 6, а зовсім інша технологія, хоча в цілому принципи роботи у них можуть співпадати. Тому в рамках EF Core використовується своя система версій.

Як технологія доступу до даних Entity Framework Core може використовуватися на різних платформах стеку .NET. Це і стандартні платформи типу Windows Forms, консольні додатки, WPF, UWP і ASP.NET Core. При цьому, ми можемо задіяти її не тільки на ОС Windows, але і на Linux і Mac OS X, бо вона є кросплатформною.

Основою Entity Framework є сутності або entity. Сутність визначає набір даних, що пов'язані з певним об'єктом. Тому дана технологія передбачає роботу не з таблицями, а з об'єктами і їх колекціями.

Будь-яка сутність, як і будь-який об'єкт з реального світу, має низку властивостей. Наприклад, якщо сутність описує людину, то ми можемо виділити такі властивості, як ім'я, прізвище, зріст, вік. Властивості необов'язково представляють прості дані типу int або string, але можуть також представляти і більш комплексні типи даних. І у кожної сутності може бути одна або кілька

властивостей, які будуть відрізняти цю сутність від інших і будуть унікально визначати цю сутність. Подібні властивості називають ключами.

При цьому сутності можуть бути пов'язані асоціативними зв'язками один-до-багатьох, один-до-одного і багато-до-багатьох, подібно до того, як в реальній базі даних відбувається зв'язок через зовнішні ключі.

Відмінною рисою Entity Framework Core, як технології ORM, є використання запитів LINQ для вибірки даних з БД. За допомогою LINQ ми можемо створювати різні запити на вибірку об'єктів, в тому числі пов'язаних різними асоціативними зв'язками. А Entity Framework при виконання запиту перетворює вираження LINQ у вирази SQL.

3.1.6 Бібліотека ML.NET

ML.NET – це безкоштовна бібліотека машинного навчання програмного забезпечення для мов програмування C# і F#. Дана бібліотека також підтримує моделі Python при використанні їх разом з NimbusML. Попередній випуск ML.NET включав в себе перетворення для таких функцій, як створення n-грамів, і навчань, які оперують бінарною класифікацією, багатокласовою класифікацією та регресійними завданнями. З тих пір були додані додаткові можливості, такі як системи виявлення аномалій та рекомендацій, а інші підходи, такі як глибоке навчання, будуть включені в наступних версіях.

Використовуючи дану бібліотеку ви можете навчати користувацьку модель, вказавши відповідний алгоритм, або використати попередньо навчені моделі створені в TensorFlow чи ONNX. Отриману модель можна використовувати в застосунках для отримання прогнозів. ML.NET є кросплатформною, тому працює на таких операційних системах: Windows, Linux і macOS з використанням .NET Core, або у Windows використовуючи при цьому .NET Framework.

За допомогою даної бібліотеки можна отримувати прогнози наступних типів:

Таблиця 3.1 – Типи підтримуваного прогнозування в ML.NET

Класифікація/категоризація	Автоматичний розподіл відгуків клієнтів на позитивні та негативні
Регресія	Прогнозування ціни на будинки, виходячи з їх розміру і місцезнаходження
Виявлення аномалій	Виявлення шахрайських банківських операцій
Рекомендації	Пропозиція продуктів, які онлайн-покупці можуть захотіти купити, на основі їхніх попередніх покупок
Часові ряди/ послідовності	Прогнози погоди і об'ємів продажу
Класифікація зображень	Класифікація патологій на медичних знімках

3.2 Реалізація серверної частини та навчання нейромережі

Оскільки з архітектурою та з технологіями визначились, можемо приступати до розроблення самої системи.

Для розробки нам знадобиться середовище, в якому ми будемо працювати. Для цих цілей я обрав Visual Studio 2019.

Далі нам знадобиться великий масив даних, а саме зображень де зображено тварин. В моєму випадку я хочу навчити мережу визначати, що на фото зображено kota. Тому йду в інтернет для того, щоб завантажити 1000 картинок, на яких зображено котів. Вдалося знайти трохи більше, ніж хотілося, але це навіть краще, бо чим більший масив тесових даних, тим краще система працюватиме в майбутньому.

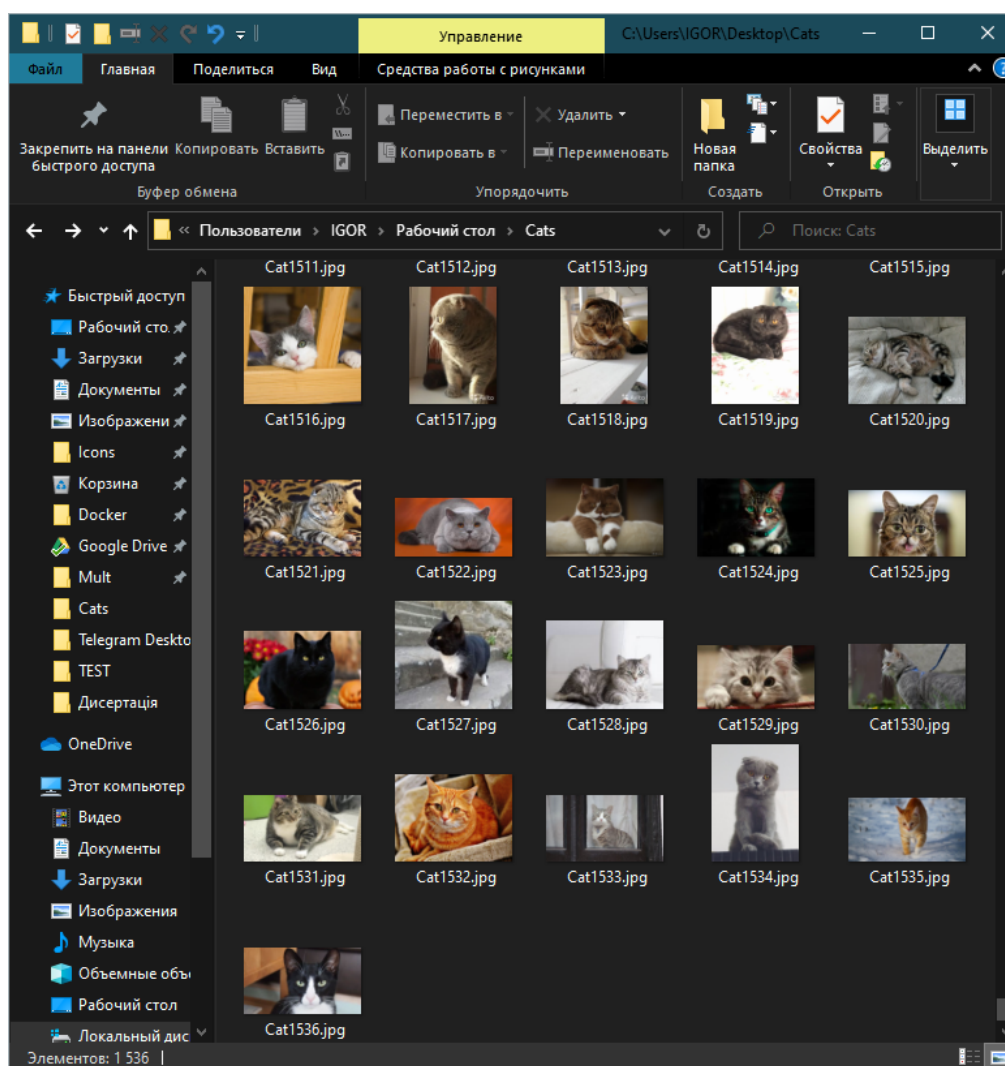


Рисунок 3.4 – Зображення котів

Далі нам потрібно позначити, що на цих всіх картинках зображено котів.
Що я і зробив.

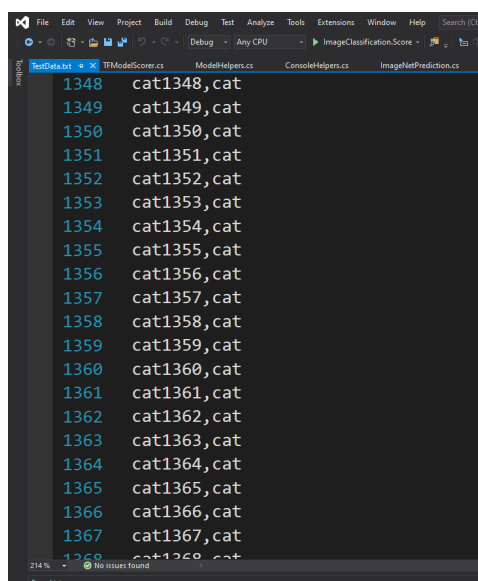


Рисунок 3.5 – Набір тестових даних для навчання нейромережі

Наступним кроком буде навчання мережі. Для цього потрібно передати наш файл з позначками до програми. Після навчання давайте протестуємо програму наскільки точно вона навчилася визначати котів на зображеннях. Візьмемо фото з інтернету.

Що ж, протестуємо, систему.

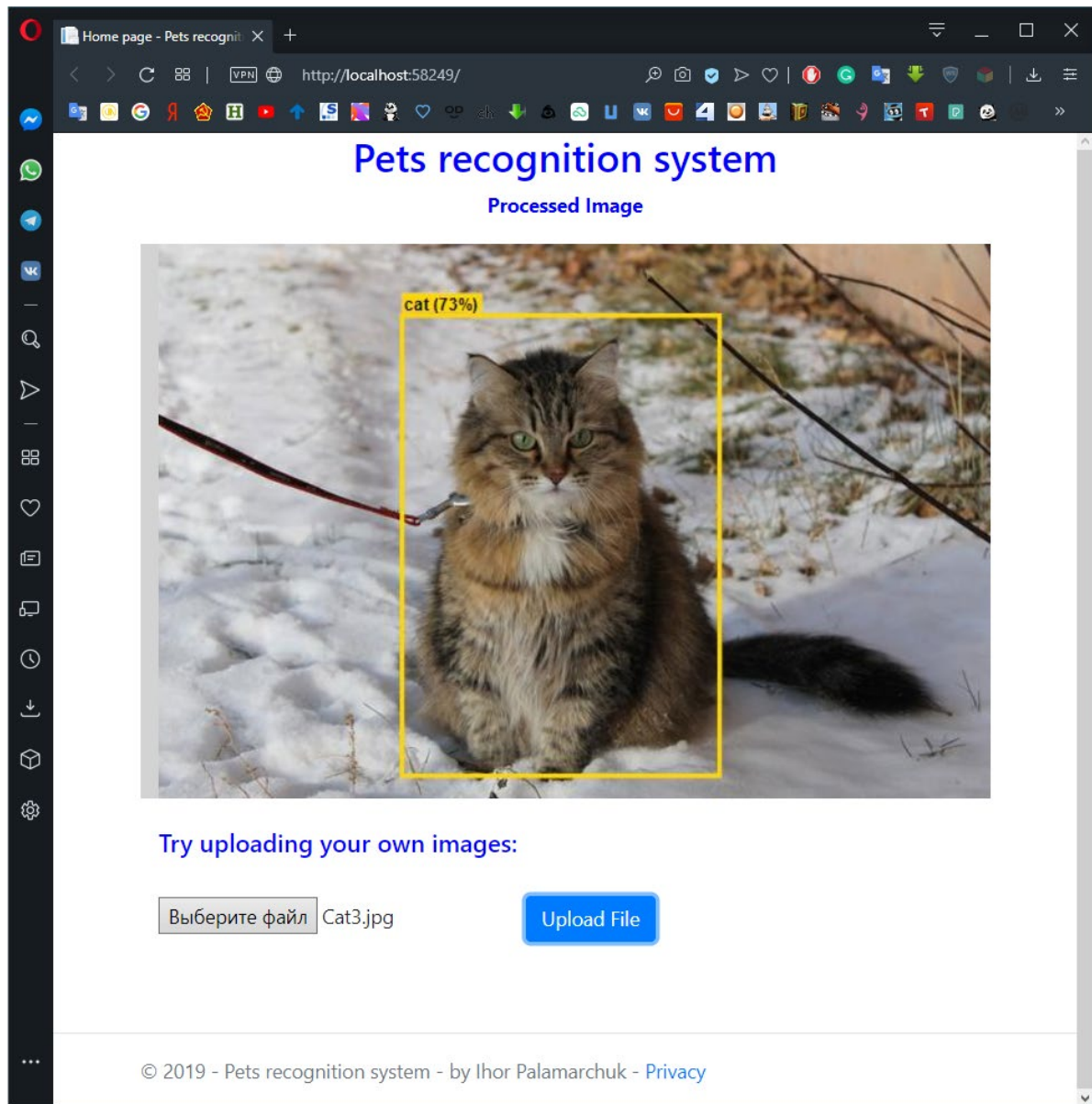


Рисунок 3.6 – Вигляд роботи системи

Як бачимо, система вважає, що це кіт. Імовірність цього становить – 73%. Далі, таким же чином навчимо визначати певного улюбленця. Для цього знову треба зібрати тестові дані та провести навчання системи. Ці дані повинні містити фотографії тільки мого улюбленця, більше нікого. Ось так виглядає мій кіт:



Рисунок 3.7 – Зображення мого kota

Давайте проведемо ще раз тестування системи. Тепер вона вже вміє визначати, чи на фото зображено саме мого домашнього улюбленця. Для початку спробуємо на ще одній фотографії з інтернету. А далі візьмемо фото мого kota для тестування, але візьмемо фото, де кіт не буде дивитися у об'єктив, щоб ускладнити задачу системі. Тож, приступимо до тестування. Результати роботи програми можна побачити на графічному додатку 4.

Як бачимо експеримент пройшов успішно. Перший раз система визначила, що на фото кіт, з імовірністю 92%, та сказала нам, що даний кіт не є моїм улюбленцем. При завантаженні фотографії мого kota, система показала, що це теж кіт, з імовірністю 93% і на цей раз, що даний кіт являється моїм домашнім улюбленцем.

3.3 Використання системи та алгоритм її роботи

Дана система буде використовуватися у системі розумного дому. Ось основні елементи, які потрібні для роботи системи:



Рисунок 3.7 – Основні елементи системи

1. Камера, що під'єднана до системи розумного дому.
2. Сервер, який встановлено в будинку, за допомогою якого ми можемо ідентифікувати нашого домашнього улюбленця.
3. Спеціальний прохід, за допомогою якого тварина зможе потрапити усередину.

Система працює за таким алгоритмом:

1. Встановлена відеокамера відправляє відеопотік на сервер.
2. Сервер отримує відео.
3. Сервер обробляє відео та розбирає його на кадри.
4. Кадр подається на нейронну мережу.
5. Мережа визначає, що зображено на кадрі.

6. Якщо це наша тварина – система за допомогою спеціального сервісу відправляє повідомлення на телефон користувача системи розумного дому.
7. Користувач бачить повідомлення, та за допомогою системи розумного дому може підключитися до відеокамери та подивитися , що це справді його домашня тварина. Коли він впевнений, що це справді вона - посилає команду до системи, яка відчиняє спеціальний прохід.
8. Домашня тварина може увійти всередину будинку.

Висновки до розділу

В даному розділі визначено архітектуру, що буде використано для створення даної системи. Системи, які побудовані на основі такої архітектури зазвичай розділені на три шари:

- Користувацький інтерфейс;
- Серверна частина;
- Шар доступу до даних (БД) .

Також визначено стек технологій, які використано для створення застосунку. Мовою програмування обрано C#, платформою на якій вестиметься розробка обрано .NET Core 3.0, MS SQL Server 2019 обрано в якості застосунку для зберігання даних до БД, для того, щоб отримати доступ до цих даних з створеної системи використано Entity Frameworks Core. ML.NET для навчання системи визначати тварин на фото.

В ході роботи було проведено навчання системи тестовими даними, та проведено експеримент, в ході якого визначили, що застосунок працює та має високий відсоток правильних відповідей.

РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

Ідея проекту полягає у створенні системи для розпізнавання домашніх тварин, для використання у системах розумного дому. Розглянемо зміст ідеї, можливі напрямки застосування. Основні переваги, які зможе отримати користувач представлено у табл. 4.1.

Таблиця 4.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Система для визначення тварин	Визначення домашніх улюбленців у системах розумного дому	Користувачі використовуючи даний застосунок мають можливість визначати чи саме їх домашній улюбленець повернувся додому, та за допомогою системи розумного дому впустити його всередину будівлі

Даний проект вирізняється тим, що допомагає спросити не тільки потреби людини, а її менших друзів - тварин. Далі проведено аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників).

Визначено перелік техніко-економічних властивостей та характеристик ідеї. На ринку наявні декілька конкурентних продуктів, які мають схожий функціонал, однак все-ж таки відрізняються за певними критеріями.

Проведено порівняльний аналіз показників: для власної ідеї визначаються показники, що мають а) гірші значення (*W*, слабкі); б) аналогічні (*N*, нейтральні) значення; в) кращі значення (*S*, сильні) (табл. 4.2)

Таблиця 4.2. Опис ідеї стартап-проекту

№	Техніко-економічні характеристики ідеї	Продукція конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Google Lens	Image Recognizer	Imagga			
1	Швидкість роботи	Швидка	Швидка	Швидка	Швидка	Безпосередня робота з базою даних	Простота архітектури	Накопичення інформації в одному місці
2	Зручність використання	Зручно	Зручно	Відсутня	Зручно	Мала кількість налаштувань	Перспективи розвитку UI	Простота наявного функціоналу
3	Вимоги до системи	Середні	Мінімальні	Середні	Середні	Відсутня оптимізація під старі системи	Оптимізована робота через веб-додаток	Актуальність програми для нових систем через використання новітніх бібліотек
4	Кросплатформенність	Наявна	Наявна	Відсутня	Наявна	Серверна частина не є кросплатформеною	Можливість розширення API для створення нового функціоналу	Повна кросплатформенність, оскільки системою можна користуватися з веб-браузера
5	Функціонал	Мало функцій	Багато функцій	Мало функцій	Достатньо функцій	Мала кількість функціоналу	Гнучке API і модель	Можливість розширювати функціонал достатньо швидко за потреби

4.2 Технологічний аудит ідеї проекту

Таблиця 4.3. Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Робота з накопиченим масивом інформації	Робота через API бази даних	База даних та API	Повністю відкрита для роботи з інформацією
2	Робота з інформацією, яку вводить користувач	API для роботи з обробки даних	Розроблені бібліотеки для роботи нейронної мережі	Повністю відкритий програмний код та використання готових бібліотек
3	Робота з тестовим масивом інформації та даними, які вводить користувач	Використання всіх вище технологій	База даних та API. Розроблені бібліотеки для роботи нейронної мережі	Повністю відкрита для роботи з інформацією. Повністю відкритий програмний код та використання готових бібліотек
<i>Обрана технологія реалізації ідеї проекту: 3</i>				

Висновок: технологічна реалізація продукту – можлива, вибрана технологія №3, яка може нам допомогти розробити якісний продукт з використанням комбінації технологій, які працюють з великим відкритим масивом даних та обробкою як наявних даних, так і нових даних, введених користувачем. Техніко-економічні характеристики на початковому етапі введення системи в експлуатацію будуть відставати від світових аналогів, які використовуються для іншого призначення та цілей.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів конкурентів. Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 4.4).

Таблиця 4.4. Попередня характеристика потенційного ринку

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн./ум.од.	25000
3	Динаміка ринку	Стагнація національної економіки. Темпи розвитку світової економіки позитивні, але з ознаками зменшення росту
4	Наявність обмежень для входу	Відсутні. Конкуренти займають свої певні сфери, які висвітлюють функції, які вказані в дисертації стартапу, тільки як додаткові
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі або по ринку, %	69

Висновок: враховуючи кількість головних гравців по ринку, зростаючу динаміку ринку, невелику кількість конкурентів та середню норму рентабельності можна зробити висновок, що на даний момент, ринок для входження стартап продукту є привабливим та надає змогу реалізувати продукцію на світовому ринку, оскільки на національному ринку економічний розвиток не надає необхідного результату для вдалої економічної реалізації проекту.

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці цільових груп клієнтів	Вимоги споживачів до товару
1	Потреба розпізнавання тварин	Дослідники	Стартап буде виконувати функції розпізнавання тварин	Зручність у використанні. Швидка робота системи. Спроможність швидко освоїти як користуватися системою. Можливість впроваджувати користувачам свій функціонал у систему для більшого розвитку проекту серед інших цільових аудиторій
2	Потреба розпізнавання тварин та можливість їх класифікувати, як домашніх улюбленців	Власники/розробники систем розумного дому	Стартап буде виконувати функції автоматичного визначення тварини та класифікуватиме її, як домашній улюбленець, якщо це так, або ж ні, якщо такою вона не являється	

Таблиця 4.6. Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренти	Наявність конкурентів котрі надають схожі рішення	Зменшення ціни на поставлену послугу; Розробка унікальних характеристик товару; Надання ліцензій на обслуговування
2	Кошти на розробку та підтримку продукту	Закінчення грошей та недостатнє фінансування	Залучення додаткових інвесторів, мотивація роботи на перспективу; Ітеративна розробка продукту задля покрокового виведення продукту на ринок та отримання відповіді користувачів
3	Вихід аналогу	Вихід аналогу даного товару може призвести до знецінення та безідейності даного товару	Вихід товару на ринок в коротші строки з не повною, але достатньою, функціональністю для зацікавлення усіх цільових аудиторій; Проведення рекламної компанії

Таблиця 4.7. Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Новий продукт	Вихід на ринок, Зменшення монополії, Надання нових рішень у сфері	Розробка нової функціональності; Вихід нової продукції на ринок; Надання різноманітних типів ліцензій в залежності від потреб користувача \ замовника
2	Вихід аналогу	Надати продукт з певними характеристиками та можливостями що відсутні у компаній конкурентів	Аналіз ринку та користувачів задля задоволення їх потреб та надання функціональності у найкоротші строки за ціну, котра є дешевшою ніж у продуктів-замінників
3	Зворотній зв'язок від користувачів	Можливість отримання необхідної інформації для вдосконалення продукту	Наявність вхідних даних та реакція на них з боку команди розробників задля задоволення потреб та бажань кінцевих користувачів системи
4	Грошова винагорода за рекламу	Можлива комерціалізація продукту на основі реклами задля отримання грошової винагороди для подальшого розвитку	Точкова комерціалізація продукту; Введення реклами; Ведення додаткових коштів у проект задля його подальшого розвитку

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Тип конкуренції: монополістична	Товар від кожної компанії на ринку, являється недосконалим замінником товару, реалізованого іншими фірмами; На ринку є умови для входу та виходу; Ціна корелює між суперниками;	Розробка продукту з характеристиками, які покривають сфери вживання що не покривають інші товари-замінники; Кореляція цін у відповідності до товарів замінників; Різні типи ліцензій

Закінчення таблиці 4.8

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
2	Рівень конкурентної боротьби: світовий	Всі продукти замітники розроблялись інтернаціональними командами з різних куточків світу, продукти не належать до певної держави, а належать команді розробників	Вихід на ринок збуту продукту з клієнто-необхідною функціональністю; Налагодження маркетингу на основних Інтернет ресурсах задля охоплення великої кількості потенційних користувачів; Надання бета-версій продукту
3	Галузева ознака: внутрішньогалузева	Даний тип продукту може використовуватися тільки у сфері розробки ІТ додатків \ продуктів	Надання зручного, інтуїтивно зрозумілого інтерфейсу; Підтримка всім відомих методів взаємодії з середовищем розробки; Наявність документації та он-лайн підтримки
4	Конкуренція за видами товарів: товарно-видова	Дана конкуренція – конкуренція між товарами одного виду	Впровадження функціональності яка відсутня у товарів-замінників; Спрощення інтерфейсів; Надання підтримки
5	Характер конкурентних переваг: цінова та не цінова	Цінові переваги – точкова комерціалізація; Не цінова – надання функціональності, що відсутня у товарах-замінниках	Надання платних ліцензій лише на критично важливу функціональність для клієнта з певним строком підтримки, що зазначена у відповідній ліцензії; Впровадження унікальної функціональності
6	За інтенсивністю: марочна	Наявність унікального знаку що відрізняє даний продукт від продуктів-замінників	Впровадження власної назви та власного знаку

Таблиця 4.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Imagga	Google Lens, Image Recognizer	Офіційні представники компаній, реселлери	Форуми, точки продажу	Частково присутні
Висновки	Прямі конкуренти намагаються сконцентруватися на інших напрямках свої продуктів.	Потенційні конкуренти мають дуже специфічну клієнтуру до якої зазвичай входять корпоративні клієнти. Конкуренція може відбутися тільки продажах на корпоративний рівень клієнтів	Постачальники диктують умови збереження даних, які захищають приватність користувачів. Також постачальники не дають змогу зловживати етичними нормами.	Клієнти можуть диктувати умови на ринку тільки через повідомлення на форумах або в полі відгуків в точках продажу додатку.	Часткові рішення присутні на ринку

Проаналізувавши можливості роботи на ринку з огляду на конкурентну ситуацію можна зробити висновок: оскільки кожний з існуючих продуктів не впливає у великій мірі на поточну ситуацію на ринку в цілому, кожний з існуючих продуктів має свою специфічну сферу використання та свої позитивні та негативні сторони щодо рішення певних типів задач, то робота та вихід на даний ринок є можливою і реалізованою задачею.

Для виходу на ринок продукт повинен мати функціонал що відсутній у продуктів-аналогів, повинен задовольняти потреби користувачів, мати необхідний та достатній функціонал з конфігурування, підтримку зі сторони розробників та можливість розробки спеціального функціоналу за відповідною ліцензією. Окрім того, потрібно забезпечити користувача адекватною технічною підтримкою та спроможністю розробникам отримувати відгуки від користувачів.

Таблиця 4.10. Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Прагматичність	Через запуск стартапу система буде не дуже складною з точки зору архітектури перший час. Через певний період із додаванням функціоналу та оптимізації алгоритмів роботи програмний код буде все складнішим. Такий етап наступить не раніше одного року постійної роботи над проектом.
2	Зручність	Оскільки стартап розробляється на багатьох платформах з різною шириною екранів, то зручність використання системи на різних пристроях буде відігравати не малу роль у спроможності конкурувати з іншими гравцями ринку
3	Швидкість роботи	Швидкість роботи відіграє велику роль для користувачів, оскільки вони не будуть готові чекати декілька хвилин на виведення результату роботи додатку.
4	Оптимізація	Якщо додаток буде дуже часто видавати помилки при роботі, то користувачі не будуть вважати додаток надійним.
5	Налаштування під користувача	Різні люди мають різні звички, які вони використовують, наприклад, якщо є люди, які люблять працювати за додатком де є темні кольори, а є такі люди, які люблять світлі кольори. Можливість редагувати зовнішній вигляд додатку надає значну перевагу серед конкурентів.
6	Відкритість вихідного коду	При наявності вихідного коду будь-який продукт має перспективи розвиватися у багатьох напрямках, особливо таких, які можуть бути неочевидні на перший погляд.
7	Приватність	В останні роки приватність людей та інформація щодо них все частіше зловживається шахраями або великими корпораціями, які потребують погодження з умовами доступу до приватної інформації та її обробки.

Закінчення таблиці 4.10.

№	Фактор конкурентоспроможності	Обґрунтування
8	Технічна підтримка	Якщо технічна підтримка компанії буде працювати своєчасно та швидко, то це допоможе зберегти репутацію компанії на відміну від конкурентів, де їй не приділяють увагу
9	Документація	Будь-який додаток, особливо якщо він має новий функціонал, повинен бути добре роз'яснений своїм користувачам та як його можна використовувати, щоб не мати проблем при подальшій роботі з ним
10	Ціна	Чим дешевше товар, тим більше шансів що його можуть купити, особливо якщо він працює краще ніж конкурентний товар

За визначеними факторами конкурентоспроможності (табл. 4.10) проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 5.11). (С.П. – стартап проект, К.1 – Google Lens, К.2 – Image Recognizer, К.3 – Imagga)

Таблиця 4.11. Порівняльний аналіз сильних та слабких сторін системи

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з запропонованим						
			-3	-2	-1	0	+1	+2	+3
1	Прагматичність			С.П.	К.2	К.3	К.1		
2	Зручність					К.2	К.1	К.3	С. П.
3	Швидкість роботи					К.1	К.2	К.3	С. П.
4	Оптимізація			К.2		К.1	К.3		С. П.
5	Налаштування під користувача				К.2	С.П	К.3	К.1	
6	Відкритість вихідного коду		К.2	К.3	С.П.	К.1			
7	Приватність				К.3	К.2	К.1	С.П	
8	Технічна підтримка			К.2	К.1	К.3			С. П
9	Документація				К.2	К.3		К.1	С. П
10	Ціна			К.3	К.2	К.1		С.П.	

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 4.10).

Таблиця 4.12. SWOT аналіз стартап-проекту

<p>Сильні сторони (S):</p> <ul style="list-style-type: none"> – Прагматичність системи через її легкість роботи; – простота у використанні; – наявність можливості розширення функціоналу; – збереження приватності інформації користувача; – швидкість роботи системи; – зручність 	<p>Слабкі сторони (W):</p> <ul style="list-style-type: none"> – Неоптимізованість алгоритму; – відкритість вихідного коду системи
<p>Можливості (O):</p> <ul style="list-style-type: none"> – Зворотній зв'язок з клієнтською базою компанії для спроможності розвивати проект в інші напрямки; – можливість створювати нові модулі системи для розширення функціоналу; – можливість зручної інтеграції цих модулів 	<p>Загрози (T):</p> <ul style="list-style-type: none"> – Можлива автономна робота алгоритмів; – складність роботи алгоритму при невиявлених випадках використання додатку

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (табл. 4.9, аналіз потенційних конкурентів). Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів (табл. 4.13).

Таблиця 4.13. Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Безкоштовне надання певного функціоналу у користування споживачам на обмежений термін	Головний ресурс – люди, даний ресурс - наявний	4-6 місяців
2	Реклама	Залучення власних коштів для реклами товару	2-3 місяці
3	Написання статей та опис товару на відомих ресурсах	Головний ресурс – час, даний ресурс - наявний	2-3 тижні
4	Презентація товару на хакатонах й інших ІТ заходах, на тематичних форумах, виставках по виробничій лінії	Ресурс – час та гроші для участі, наявні	3-6 місяці

Основною альтернативою є презентація товару на хакатонах й інших ІТ заходах, на тематичних форумах, виставках по виробничій лінії.

4.4 Розроблення ринкової стратегії проекту

Таблиця 4.14. Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Компанії, що розробляють системи розумного дому	Присутня	Середній	Низька	Середня
2	Користувачі систем розумного дому	Присутня	Високий	Низька	Легка
Які цільові групи обрано: 1, 2					

Відповідно до проведеного аналізу можна зробити висновок, що підходящою цільовою групою для розповсюдження даного програмного продукту є компанії, які спеціалізуються на випуску програмного забезпечення для систем розумного дому та їх потенційні користувачі. Відповідно до стратегії охоплення ринку збуту товару обрано стратегію масового маркетингу, оскільки для цільових груп в цілому надається стандартизований продукт з можливістю розширення функціональності за домовленістю (відповідно до ліцензії). Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (табл. 4.15).

Таблиця 4.15. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Надання функціональності що відсутня у товарів-замінників, підтримка клієнтів	Проведення реклами, освітлення унікальної функціональності через інтернет ресурси та інші канали, контакт напряду з споживачами; формування лояльності і прихильності споживачів	Зниження ступеню замінності товару; Прихильність клієнтів; Відмітні властивості товару; Відмітні характеристики товару;	Стратегія диференціації

Таблиця 4.16. Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки
Так, буде	Так, ціль компанії знайти нових споживачів та, частково, забрати існуючих у конкурентів задля задоволення потреб останніх	-	Стратегія заняття конкурентної ніші

Таблиця 4.17. Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Зручність	Диференціація	Спроможність швидше проводити складські процеси	Спроможність економити час на пошук нової локації
2	Алгоритми роботи	Заняття конкурентної ніші	Перспектива розвитку проекту	Розвиток в науці
3	Документація	Диференціація	Зручність користування системою через відкриті навчальні статті та документацію	Зручність у користуванні

Відповідно до проведеного аналізу можна зробити висновок, що стартап-компанія вибирає як базову стратегію розвитку – стратегію диференціації, як базову стратегію конкурентної поведінки – стратегію заняття конкурентної ніші.

4.5 Розроблення маркетингової програми стартап-проекту

Таблиця 4.18. Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Розпізнавання тварини	Вирішує потребу	Зручність та швидкість роботи
2	Класифікація розпізнаної тварини, як домашнього улюбленця	Автоматичне виконання даної операції	Повністю інтелектуальне та автоматичне виконання розпізнавання та класифікація тварин

Таблиця 4.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
1. Товар за задумом	Система розпізнавання домашніх тварин		
2. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Зручність	Нм	Е
	Швидкість роботи	Нм	Тх
	Оптимізація	Нм	Тх
	Ціна	Нм	Е
	Документація	Нм	Тл
	Технічна підтримка	Нм	Тх
	Приватність	Нм	Тз
	Налаштування під користувача	Нм	Ор
	Функціональність	Нм	Е
3. Товар із підкріпленням	До продажу: наявна повна документація, акції на придбання декількох ліцензій, знижки для певних сегментів на покупку товару		
	Після продажу: додаткова підтримка спеціалістів налаштування, підтримка з боку розробника		
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності, патент			

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари аналоги або товари субституту, а також аналіз рівня доходів цільової групи споживачів (табл. 4.20). Аналіз проводиться експертним методом.

Таблиця 4.20. Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
50 тис. грн	75 тис. грн	280 тис. грн	25-70 тис. грн

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 4.21):

- проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту);
- вибір та обґрунтування оптимальної глибини каналу збуту;
- вибір та обґрунтування виду посередників.

Таблиця 4.21. Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Вибір послуг на сайті, оплата, постачання послуг	-	Виробник-споживач	Web-сайт

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Таблиця 4.22. Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Знають, які саме послуги треба для вирішення задач	Веб-сайт, телефон, зустрічі	Підтримка клієнтів, індивідуальний підхід	Донесення переваг до клієнтів	Допомога у виборі бібліотеки машинного навчання

Як результат було створено ринкову (маркетингову) програму, що включає в себе визначення ключових переваг концепції потенційного товару, опис моделі товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

Висновки до розділу

1. Описано стратегії та підходи з розроблення стартап-проекту;
2. Визначено наявність попиту, динаміку та рентабельність роботи ринку;
3. Виявлено, що існує можливість ринкової комерціалізації проекту;
4. В якості базової стратегії, обрано стратегію диференціації;
5. Кількість конкурентів націлених на потенційних клієнтів низька;
6. Наступна розробка проекту є доцільною.

ВИСНОВКИ

Нейронні мережі являють собою систему, яка може допомогти вирішити багато питань, які раніше могли вирішуватися тільки людиною. До таких можна віднести випадки розпізнавання об'єктів на зображенні або спрогнозувати погоду, курс валют, соціальну поведінку.

У ході виконання магістерської дисертації розглянуто питання пов'язані з необхідністю та актуальністю створення системи розпізнавання домашніх тварин у системі розумного дому. Наведено характеристику предметного середовища та обґрунтовано причину розробки системи. Описано вимоги до додатку даного виду та технології для створення системи.

На основі даних, отриманих в процесі аналізу, сформульовано задачу створення системи розпізнавання домашніх тварин для систем розумного дому на основі трирівневої архітектури з використанням технологій машинного навчання.

Для розробки даної системи використано мову програмування C#, середовищем для розробки обрано Visual Studio 2019, MS SQL Server 2019 для зберігання даних, Entity Framework Core для легкого та простого управління даними, що знаходяться в БД. Також використано бібліотеку ML.NET для навчання нейромережі визначати тварин на фото.

Наведено приклади використання створеного додатку. Розглянуто основний функціонал та способи його використання. Проведено збір тестових даних, а саме картинок котів у кількості, більше ніж 1500 шт. Проведено експеримент навчання нейромережі тестовими даними. Система визначає тварин з правильністю результату до 95%. Також систему навчено визначати чи являється тварина на фото, моїм домашнім улюбленцем.

Проведено маркетинговий аналіз стартап-проекту. Проведено опис ідеї проекту, технологічний аудит ідеї проекту, аналіз ринкових можливостей запуску стартап-проекту. Розроблено ринкову стратегію проекту та маркетингову програму стартап-проекту.

ПЕРЕЛІК ПОСИЛАНЬ

1. Крисилев В.А. Представление исходных данных в задачах нейросетевого программирования / Одесса: ОНПУ. 2003.
2. Градосельская Г.В. Сетевые измерения в социологии: Учебное пособие / М.: Издательский дом «Новый учебник», 2004. 248 с.
3. База и генератор образовательных ресурсов [Электронный ресурс] // МГТУ им. Н.Э. Баумана, кафедра САПР. 2003-2015. Режим доступа: <http://bigor.bmstu.ru/?cnt/?doc=NN/base.cou>
4. Шитиков В.К., Розенберг Г.С., Зинченко Т.Д. Методы системной идентификации / Тольятти: ИЭВБ РАН. 2003. 463 с.
5. Введение в теорию нейронных сетей [Электронный ресурс] // Основные понятия нейросетей. Режим доступа: <http://www.orc.ru>
6. Шахнов В.А., Власов А.И., Кузнецов А.С. Нейрокомпьютеры: архитектура и схемотехника / М.: Изд-во Машиностроение. 2000. 64 с.
7. Хайкин С. Нейронные сети. Полный курс / М.: Вильямс, 2006.
8. Халафян А.А. Статистический анализ данных. 3-е изд. учеб. / Бином – Пресс. 2007. 512 с.
9. Остроух А.В. Интеллектуальные системы Учебное пособие / Научноинновационный центр. 2015 г
10. Рассел С., Норвиг П. Искусственный интеллект: современный подход / Изд-во: Вильямс. 2006. 1424 с.
11. Rosenblatt F. 1962. Principles of Neurodynamics. New York: Spartan Books. (Русский перевод: Розенблатт Ф. Принципы нейродинамики. – М: Мир. – 1965.)
12. Grossberg S. 1969. Some networks that can learn, remember and reproduce any number of complicated space-time patterns. Journal of Mathematics and Mechanics, 19:53-91.
13. Rosenblatt, Frank. “The perceptron: a probabilistic model for information storage and organization in the brain.” Psychological review 65.6 (1958): 386.

14. Hopfield, John J. "Neural networks and physical systems with emergent collective computational abilities." *Proceedings of the national academy of sciences* 79.8 (1982): 2554-2558.
15. Hayes, Brian. "First links in the Markov chain." *American Scientist* 101.2 (2013): 252.
16. Hinton, Geoffrey E., and Terrence J. Sejnowski. "Learning and relearning in Boltzmann machines." *Parallel distributed processing: Explorations in the microstructure of cognition* 1 (1986): 282-317.
17. LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278-2324.
18. Розроблення стартап-проекту [Електронний ресурс] : Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016. – 28 с.
19. C# 8.0 and .NET Core 3.0 – Modern Cross-Platform Development: book / Mark J. Price — Packt Publishing Ltd, 2019 — 818 p.
20. Automatic Generation of Neural Network Architecture Using Evolutionary Computation: science book / E. Vonk, L. C. Jain, Ray P. Johnson — World Scientific, 1997 — 182 p.
21. Паламарчук І.О., Базака Ю.А. Система розпізнавання домашніх тварин для розумного дому. Вчені записки Таврійського Національного Університету Імені В.І. Вернадського. Том 30 (69) №6 2019.

ДОДАТОК А
Проходження перевірки роботи на співпадіння

ДОДАТОК Б

Вихідні коди класу контролера «ObjectDetectionController»

```

namespace PetsRecognition.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class ObjectDetectionController : ControllerBase
    {
        private readonly string _imagesTmpFolder;

        private readonly ILogger<ObjectDetectionController> _logger;
        private readonly IObjectDetectionService _objectDetectionService;

        private string base64String = string.Empty;
        public ObjectDetectionController(IObjectDetectionService ObjectDetectionService,
        ILogger<ObjectDetectionController> logger, IImageFileWriter imageWriter)
        {
            //Get injected dependencies
            _objectDetectionService = ObjectDetectionService;
            _logger = logger;
            _imagesTmpFolder = CommonHelpers.GetAbsolutePath(@"../../ImagesTemp");
        }
        public class Result
        {
            public string imageString { get; set; }
        }
        [HttpGet()]
        public IActionResult Get([FromQuery]string url)
        {
            string imageFileRelativePath = @"../../assets" + url;
            string imageFilePath = CommonHelpers.GetAbsolutePath(imageFileRelativePath);
            try
            {
                Image image = Image.FromFile(imageFilePath);
                //Convert to Bitmap
                Bitmap bitmapImage = (Bitmap)image;

                //Set the specific image data into the ImageInputData type used in the
                ImageInputData imageInputData = new ImageInputData { Image = bitmapImage };

                //Detect the objects in the image
                var result = DetectAndPaintImage(imageInputData, imageFilePath, "");
                return Ok(result);
            }
            catch (Exception e)
            {
                _logger.LogInformation("Error is: " + e.Message);
                return BadRequest();
            }
        }
        [HttpPost]
        [ProducesResponseType(200)]
        [ProducesResponseType(400)]
        [Route("IdentifyObjects")]
        public async Task<IActionResult> IdentifyObjects(IFormFile imageFile)
        {
            if (imageFile.Length == 0)
                return BadRequest();
            try
            {
                MemoryStream imageMemoryStream = new MemoryStream();

```

```

        await imageFile.CopyToAsync(imageMemoryStream);

        //Check that the image is valid
        byte[] imageData = imageMemoryStream.ToArray();
        if (!imageData.IsValidImage())
            return StatusCode(StatusCodes.Status415UnsupportedMediaType);

        //Convert to Image
        Image image = Image.FromStream(imageMemoryStream);

        string fileName = string.Format("{0}.Jpeg", image.GetHashCode());
        string imageFilePath = Path.Combine(_imagesTmpFolder, fileName);
        //save image to a path
        image.Save(imageFilePath, ImageFormat.Jpeg);

        //Convert to Bitmap
        Bitmap bitmapImage = (Bitmap)image;

        _logger.LogInformation($"Start processing image...");

        //Measure execution time
        var watch = System.Diagnostics.Stopwatch.StartNew();

        //Set the specific image data into the ImageInputData type used in the
        ImageInputData imageInputData = new ImageInputData { Image = bitmapImage };

        //Detect the objects in the image

        var result = DetectAndPaintImage(imageInputData, imageFilePath,
imageFile.FileName);

        //Stop measuring time
        watch.Stop();
        var elapsedMs = watch.ElapsedMilliseconds;
        _logger.LogInformation($"Image processed in {elapsedMs} miliseconds");
        return Ok(result);
    }
    catch (Exception e)
    {
        _logger.LogInformation("Error is: " + e.Message);
        return BadRequest();
    }
}

private Result DetectAndPaintImage(ImageInputData imageInputData, string
imageFilePath, string fileName)
{
    //Predict the objects in the image
    _objectDetectionService.DetectObjectsUsingModel(imageInputData);
    var img = _objectDetectionService.DrawBoundingBox(imageFilePath, fileName);

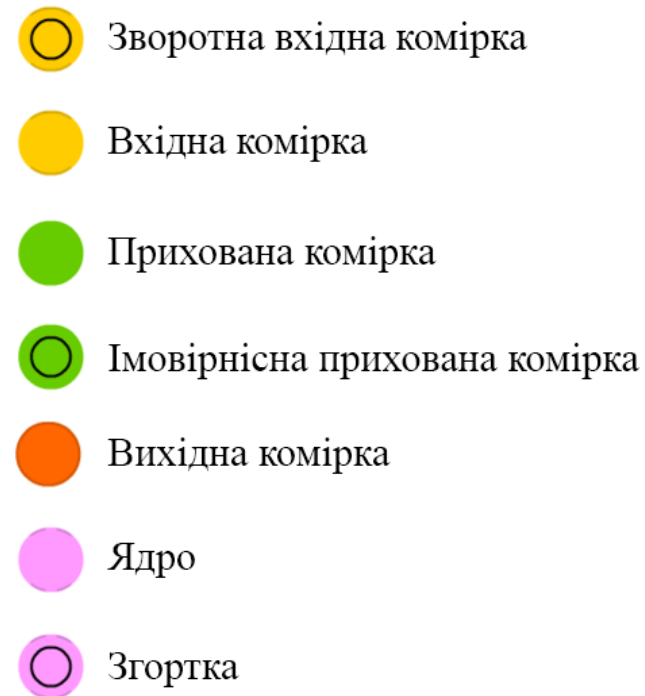
    using (MemoryStream m = new MemoryStream())
    {
        img.Save(m, img.RawFormat);
        byte[] imageBytes = m.ToArray();

        // Convert byte[] to Base64 String
        base64String = Convert.ToBase64String(imageBytes);
        var result = new Result { imageString = base64String };
        return result;
    }
}
}

```

ДОДАТОК В
Графічний матеріал, плакати

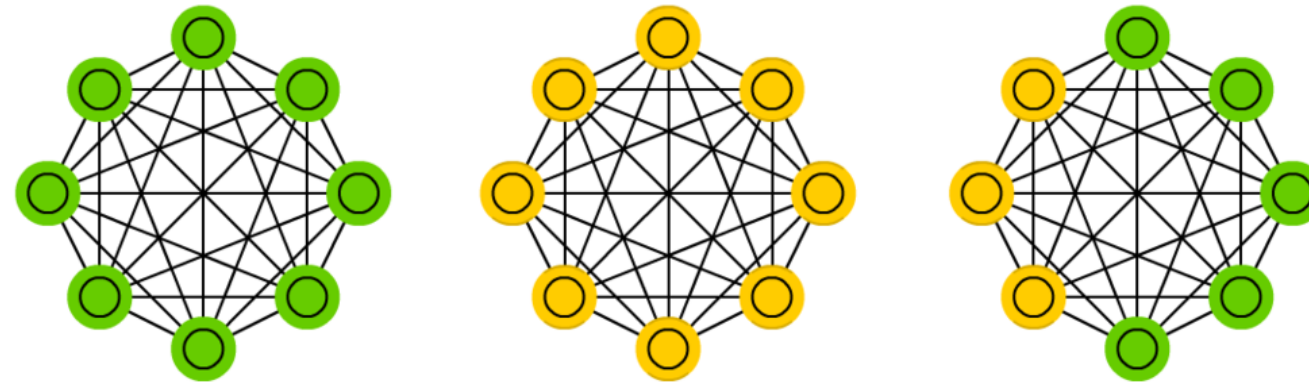
Типи нейронних мереж



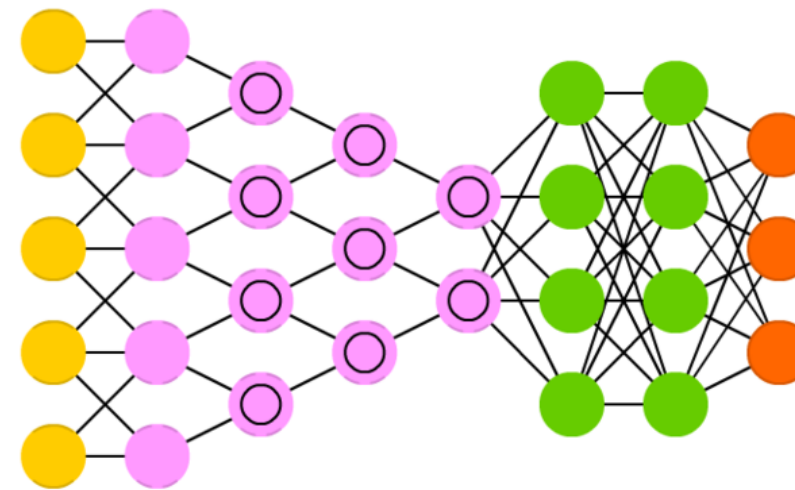
Перцептрон (P) Мережа прямого поширення (FF)



Ланцюги Маркова (MC) Мережа Гопфілда (HN) Машина Больцмана (BM)



Згорткова нейронна мережа (DCN)

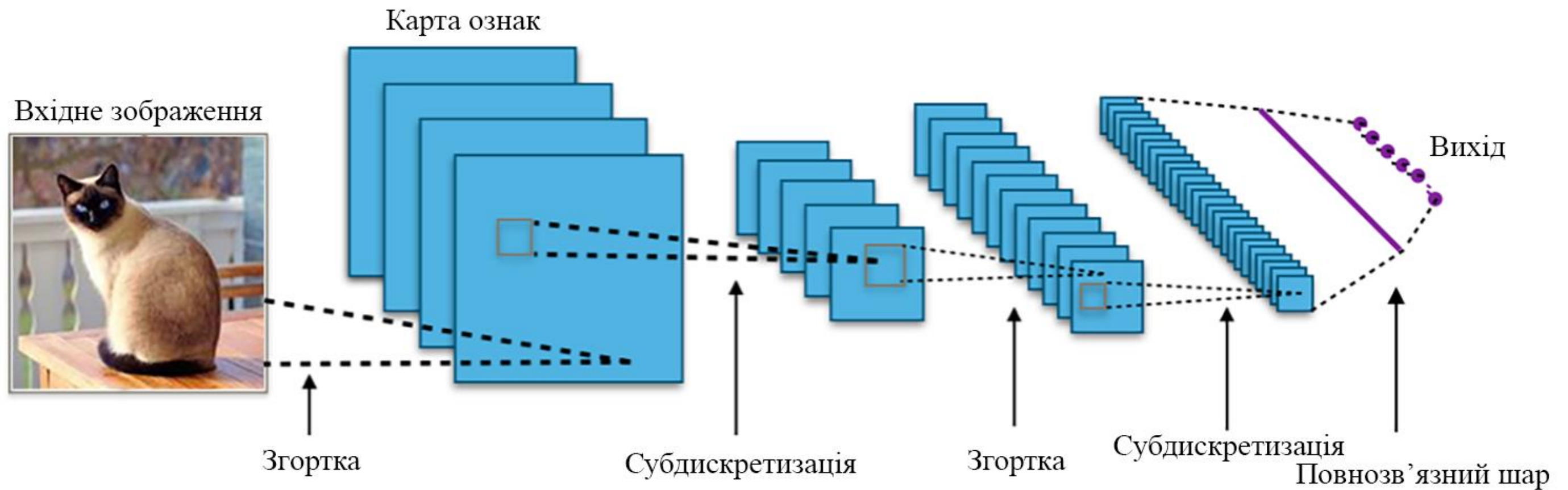


Демонстраційний плакат №_1_
до дипломної роботи на тему
„_Система розпізнавання домашніх тварин для розумного дому_”

Розробив: Паламарчук І. О.

Прийняв: _____

Концептуальне представлення згорткової мережі для класифікації зображень

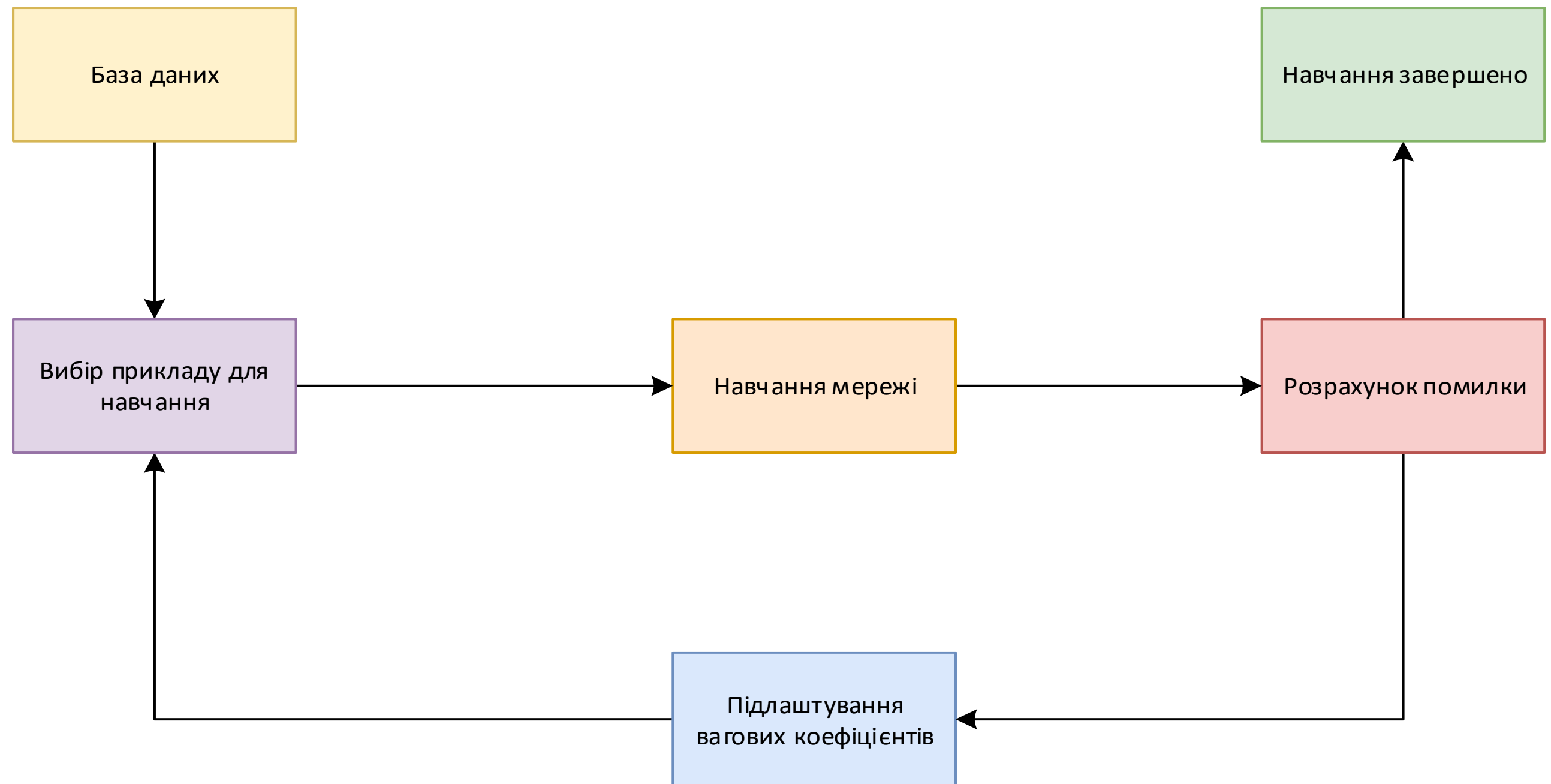


Демонстраційний плакат №_2_
до дипломної роботи на тему
„_Система розпізнавання домашніх тварин для розумного дому_”

Розробив: Паламарчук І. О.

Прийняв: _____

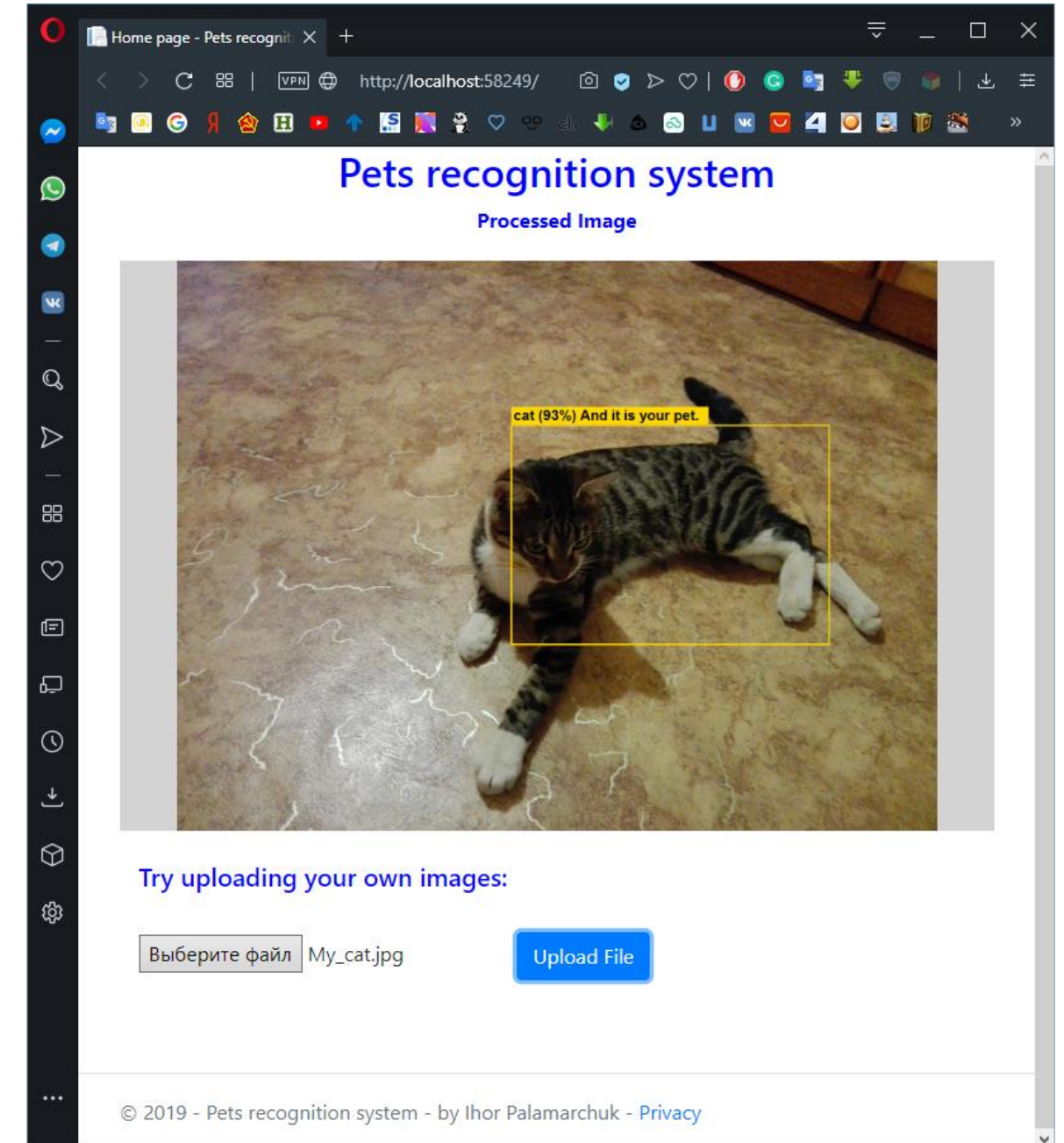
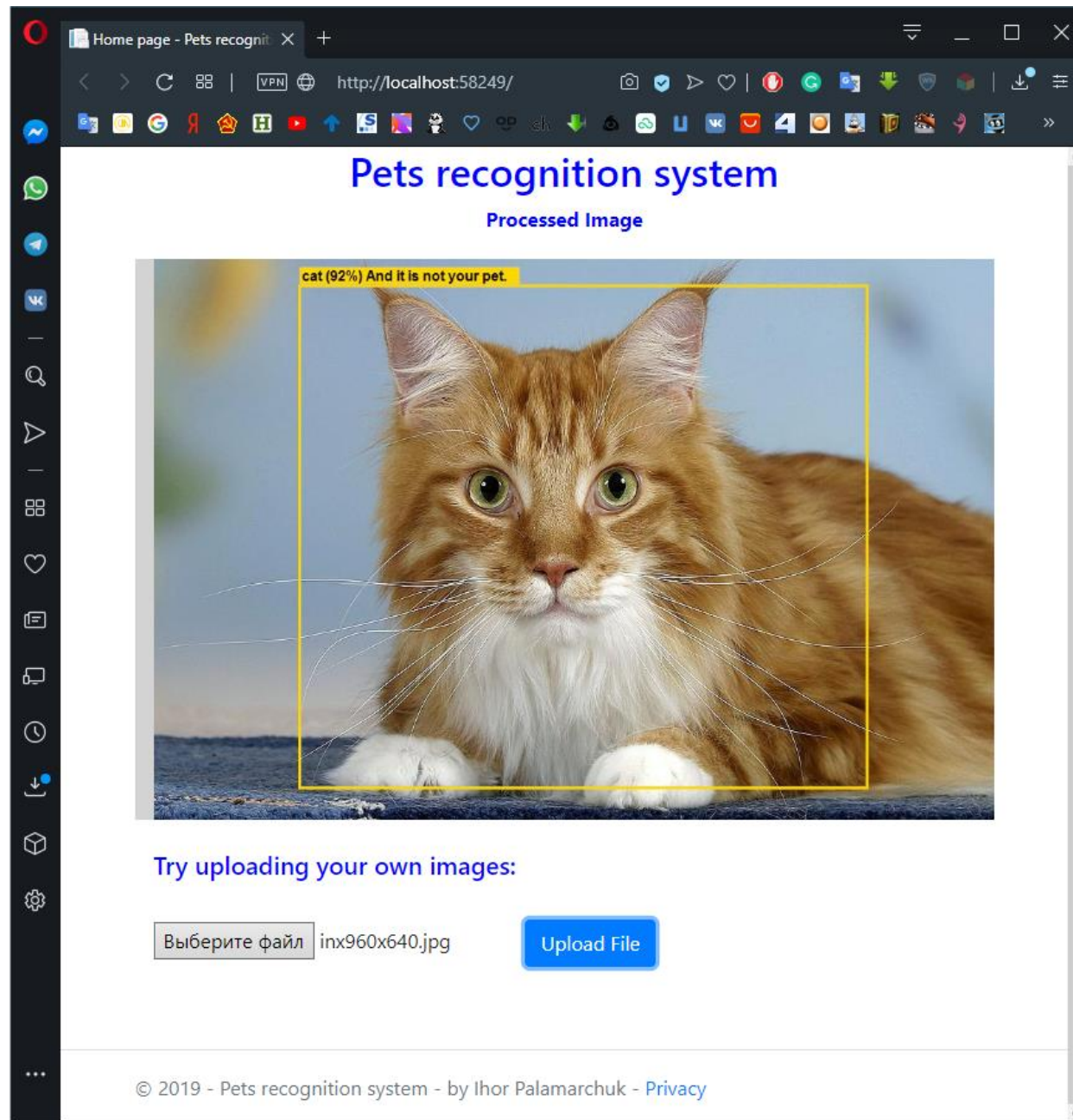
Схема навчання нейромережі



Демонстраційний плакат №_3_
до дипломної роботи на тему
„_Система розпізнавання домашніх тварин для розумного дому_”

Розробив: Паламарчук І. О.
Прийняв: _____

Результати роботи програми

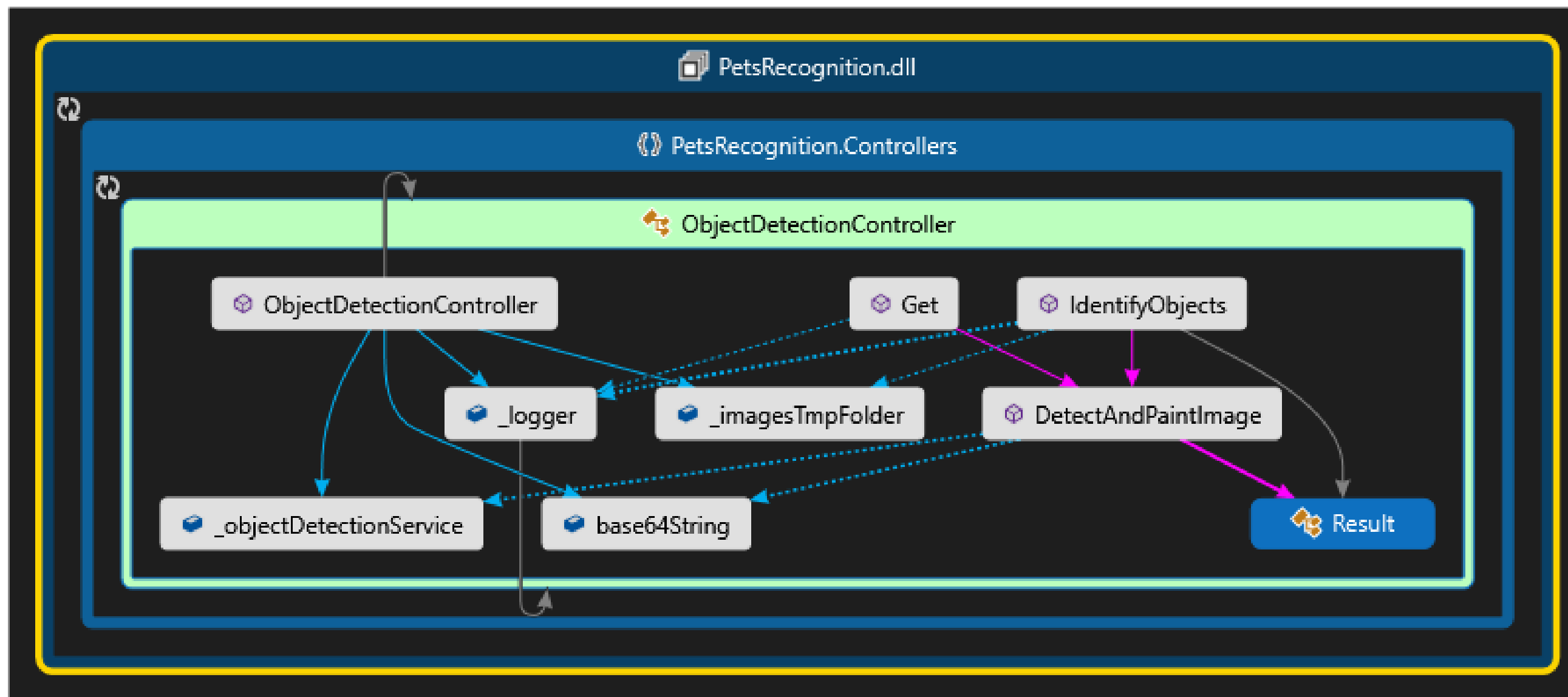


Демонстраційний плакат №_4_
до дипломної роботи на тему
„_Система розпізнавання домашніх тварин для розумного дому_”

Розробив: Паламарчук І. О.

Прийняв: _____

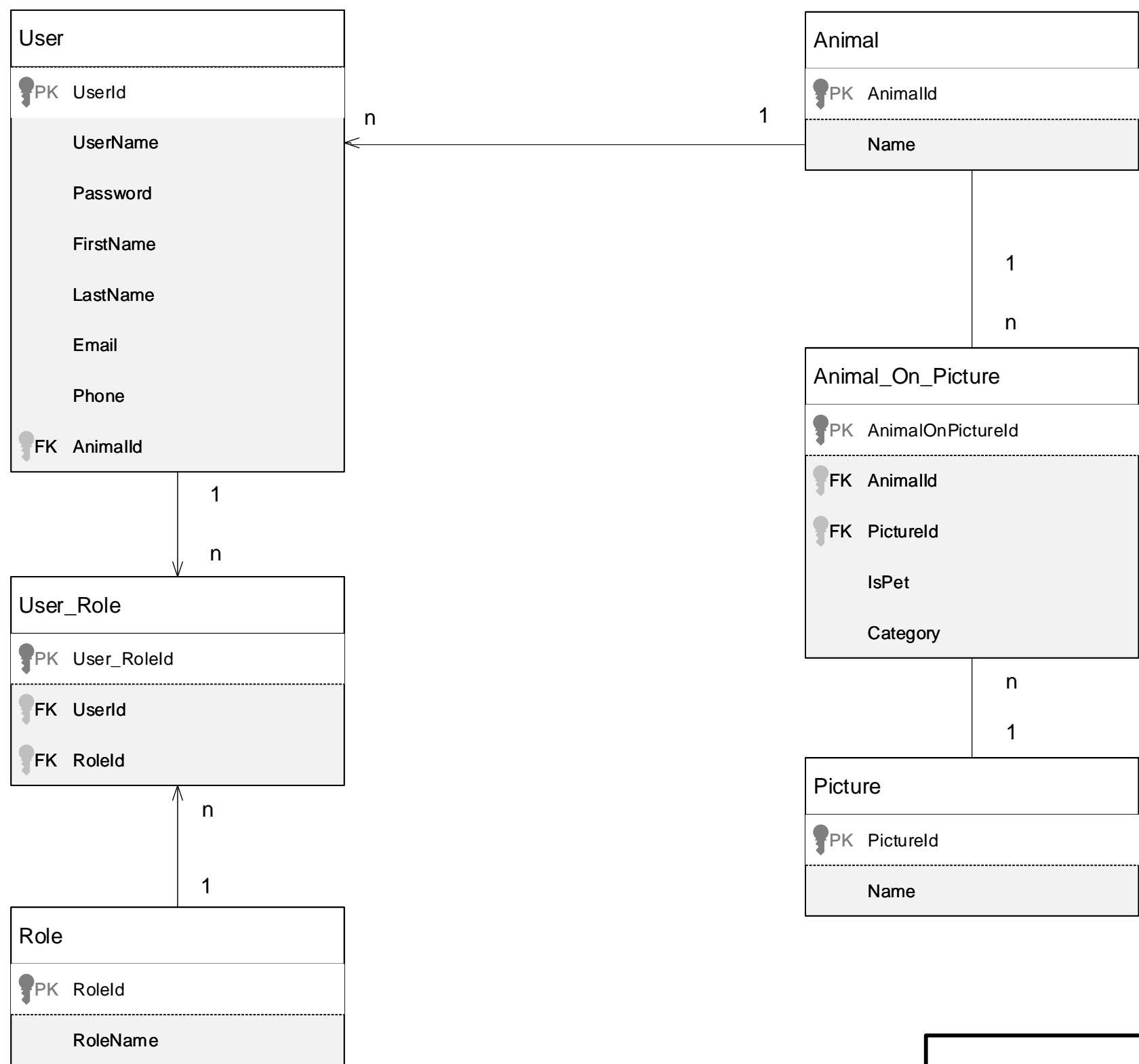
Діаграма класів



Демонстраційний плакат № 5
до дипломної роботи на тему
„Система розпізнавання домашніх тварин для розумного дому”

Розробив: Паламарчук І. О.
Прийняв: _____

Схема бази даних системи



Демонстраційний плакат №_6_
до дипломної роботи на тему
„_Система розпізнавання домашніх тварин для розумного дому_”

Розробив: Паламарчук І. О.
Прийняв: _____